Roxen CMS 5.4

Content Editor Manual

2014-03-11
www.roxen.com

Roxen Internet Software
Box 449
SE-581 05 Linköping
Sweden
www.roxen.com

Every effort has been made to ensure that the information in this document is accurate. Roxen Internet Software is not responsible for printing or clerical errors.

Other company and product names mentioned herein are trademarks of their respective companies. Roxen Internet Software assumes no responsibility with regard to the performance or use of these products.

Contents

1 Introduction

This manual covers the *SiteBuilder Editor*. This interface is available for users with more advanced needs, and requires a better understanding of the Roxen CMS repository and tools. Several features are available in the SiteBuilder Editor that are not available in the Insite Editor.

Note!

> Roxen CMS is a highly customizable product. The example images used in this manual depicts the standard look of pages and interfaces. Your installed version of Roxen CMS may differ in design, but the functionality of buttons and interfaces are the same.

Roxen CMS is a modular product, and you may not have every available feature installed on your particular site. The editor features that may or may not be available are:

- Workflow
- Advanced Version Control
- Workgroup Staging
- Multi Language Support
- LogView Analyzer
- Forum component
- Poll component
- Categories

2 Content Editor

This part focuses on editing pages in the Roxen CMS SiteBuilder and the particular systems used, such as the Access Control system or how to handle multiple versions of the same document.

It consists of three parts:

- This, the introductory section, will give an overview of the working mechanics and explain the main concepts used when editing files in a SiteBuilder environment.
- The main part describes all the procedures needed for working with SiteBuilder, with examples and screen shots.
- Finally, LogView is described in separate chapters covering the details specific to these subsystems.

Note!

Some of the functionality described in this part of the manual are optional modules, and may not be available on your particular site. These functions are:

- Workflow
- Advanced Version Control
- Workgroup Staging
- Multi Language Support
- LogView Analyzer

2.1 What is Roxen SiteBuilder?

SiteBuilder is a system for keeping track of web site file systems and users. The files are managed using the SiteBuilder Editor which includes solutions for all necessary actions of a normal file system and a Version Control system to keep track of all changes made to a file by different users. The users are managed using the built-in Access Control system which cooperates with all Roxen components.

2.2 Version Control

SiteBuilder allows several users to work on the same site and the same documents at the same time. This could easily cause problems, when someone edits a document only to find out that in the meantime, someone else has also made changes to the same document, maybe even in the same places. In SiteBuilder these problems are solved by the Version Control system.

The Version Control system is a vital part of SiteBuilder and understanding it is essential for a user of SiteBuilder. To begin with, SiteBuilder has a central repository where the live documents, i.e. the documents that will actually be sent when someone looks at the site, are stored. The history of each document is also stored: What changes have been made to it, by which user, and when.

This enables a user to change a page back to an earlier version of itself, undoing the changes made since then, which means that a mistake by a page editor is never likely to be fatal - it is always possible to revert to an earlier, working version.

SiteBuilder keeps track of who has written each part of a document, and can automatically annotate each document with its authors or log its change history.

2.2.1 Edit Area

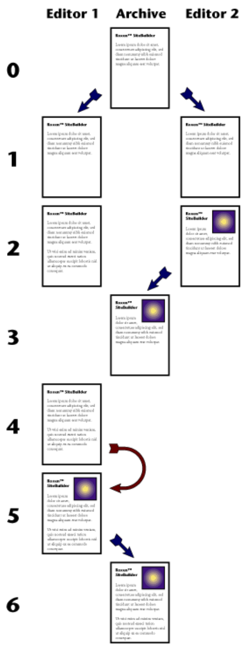
It is not possible, or desirable, to let anyone edit the live documents directly. Instead, as soon as a user wants to edit or download a document, a copy is made to a personal edit area, where you can edit the document, view it as it would look like on the site, try out different templates, illustrations or whatever. When satisfied with the result, commit the file, thus replacing the copy in the repository with the copy from the personal edit area.

2.2.2 Conflicts

Now, if somebody else is also editing the same document, it might happen that the document in the central repository has already changed. In this case, SiteBuilder will inform you that the document has changed and that you need to update your edit area to reflect the new changes. When updating, SiteBuilder will attempt to merge the changed document with your changes. Sometimes this will be simple, such as when the changes are in different parts of a long document.

Sometimes, however, there will be a collision, when both you and the other user have been editing the same paragraph. In such a case, both your and the other user's version will be shown, and you will have to decide which one to keep. Remember, it is always possible to revert to an earlier version. Finally, when you have successfully eliminated all conflicts between the document in the repository and your changes, you may commit your new document to the repository.

2.2.3 Schematic Example



Version control

A short example to show how this works:

- A document is in the central repository that contains some text. Editor1 and Editor2 start to edit it and the document is copied to each user's edit area.

- Editor1 adds some text to the document while Editor2 inserts a picture. Both users work in their own edit area and are therefore not affected by each other's changes.
- Editor2 commits her change, thereby updating the document in the repository. Since no one has changed the document in the repository since Editor2 started editing she has no problems committing.
- Editor1 tries to commit her change, but this fails since Editor2 has updated the document in the repository.
- Editor1 uses the Update wizard, which updates her document with the changes made since she started editing.
- Editor1 can now successfully commit her changes to the repository.

Note!

> The document can be reverted at a later date. Thus the version of the document that existed in step three or step one can be recovered.

2.3 Templates

One of the features of SiteBuilder is the use of templates to ensure a consistent layout and graphic profile and to assist in having a good navigation system throughout the site. Also, the template lets the content editor focus on the content he is writing, instead of worrying about HTML and layout.

2.3.1 Templates in Use

Usually all pages of the same kind, such as product presentation pages or intranet document pages, should use the same template. This means that the pages get a consistent layout and feel, of course, but it also means that it will be easy to make site-wide changes to that layout. Adding an extra link to the Christmas Special Offer to every page on the site is as easy as putting it into the framework of the products template, instead of having to edit the page for every product the company offers.

Additionally, the template may supply the content writers with extra tags. An intranet template may supply its writers with tags for adding the company logo to the text or tags to mark a word for inclusion in a site wide index page.

2.3.2 Schematic Example

The template system, based on the Extensible Stylesheet Language Transformations (XSLT) works together with the other systems of SiteBuilder. This schematic picture may illustrate the way from content file to the page presented to the user.

1
2
roxen
SiteBuilder
3
roxen
SiteBuilder
4
roxen
SiteBuilder

Templates

- A content file, consisting of some text and a picture.
- The content file together with a template. The template added some color and a headline.
- The next step is to add a navigation menu.
- Finally, the access control system is invoked, and a few of the menu choices are removed since the requesting user does not have permission to access them.

This is all done internally in SiteBuilder. Now, after all the systems of SiteBuilder have been working on the page, it is ready to be sent to the requesting user.

2.4 Access Control

The Access Control system in SiteBuilder ensures that users or visitors will not see pages that they do not have permission to access. This is done transparently, and also applies to which options are available to the users and editors at any given time.

For visitors, this means that no menu items or search results linking to pages the visitor is not allowed to access will be shown.

For users of SiteBuilder, this means that the view of the SiteBuilder interface changes depending on where it is focused. When focusing on a directory where you are permitted to write and edit files, you will have access to buttons for editing, committing, discarding and so on. When focusing somewhere where you are not allowed to change things (maybe when looking in the template directory, if you are not working with template programming) those buttons will not be available any more.

2.4.1 Groups and Permissions

So, how does this work? Every user is assigned to one or more groups by the system administrator, and every group is then assigned permissions to the different directories of the site.

For instance, you may be a writer and graphic artist, and are therefore assigned to the groups Everyone, Editors and Artists. The first group is a group where everybody is a member, and which has permission to look at only the most public parts of the site, such as the pages that are open to external visitors. The group Editors has permission to access the SiteBuilder Editor, and to write in the directories that contains the source files for the web pages. Artists has permission to write in the directories that contain images.

Your co-worker may be a member of the group TemplateCoders too, in which case he will also be allowed to write in the directory that contains the site templates. On the other hand, he might not be a member of Artists, in which case he might look around in the image directories but are not allowed to make any changes.

Your group memberships determine where on the site you are allowed to change things, which parts you are allowed to look at but not change, and which parts you are not shown at all. As a comparison, an external visitor would probably just be a member of Everyone and as such would only be able to reach the public parts of the site. Links to the SiteBuilder Editor would not be shown at all.

2.4.2 User management

Most of the management of users and permissions are handled by the system administrator, but there is one thing that the user can handle herself, and that is keeping track of and changing her password. This is done by selecting the tab Configuration and clicking Change password.

2.5 RXML

A typical web page consists of text and HTML tags, sent over the Internet from a web server to a browser, such as Mozilla Firefox or Internet Explorer. The HTML tags tell the browser how the page should be displayed. There are a lot of different browsers and versions of browsers in use, and obviously the set of HTML tags must be rather fixed and agreed upon by all. Despite this, Roxen CMS offers its own tags, called RXML tags, to extend the sometimes quite limited power of HTML, and it even allows users to define their own tags. How is this possible?

2.5.1 Macro Expansion

The answer is that the pages are not sent to the browser as they are written. Rather, the pages are processed by Roxen CMS and every instance of an RXML tag is substituted by its HTML equivalent. For instance, the `<tablify>` tag, which takes a list of things and turn them into a rather neat table, is substituted by a rather complicated expression using the ordinary HTML tag `<table>`. The `<gtext>` tag takes text and renders it as an image with selectable fonts, colors, shading, stippling and other graphical effects. When Roxen CMS reaches a `<gtext>` tag, it generates the necessary image and substitutes an ordinary HTML `<img>` tag with a link to the newly generated image.

The same happens with tags defined by the user. Whenever a user-defined tag is encountered, it is substituted with the contents of its definition. This in turn may contain other RXML tags, which are then expanded into their definition, and so on, until there are only ordinary HTML tags left.

So, the RXML tags never reach the browsers. Instead, before sending the page to the requesting browser, the server expands all RXML tags in the document to something the browser can handle.

2.6 Editing

In SiteBuilder, the easiest way to edit a file is to focus on it and selecting Edit contents from the Edit menu. If SiteBuilder is configured correctly the file will be opened in your favorite editor. Which editor to use is configured for each user and each file type.

2.6.1 Local Editor

When using a local editor any program on your computer can be used as an editor. Thus each user can choose the program(s) most suitable for her. Any type of file can be edited, for example HTML files, images and Word documents. The local editor is started by a small program, Roxen Application Launcher that must be installed first.

The local editor support must always be initiated from SiteBuilder, by selecting Edit. From within the editor program Save can be used, but not Save as nor Open. It is possible to save any number of times, until the version control functions of SiteBuilder are used (Update, Commit or Discard). Once one of the version control functions has been used it is only possible to continue editing with a local editor by pressing the Edit button again.

The reason for these limitations is that the link between SiteBuilder and the local editor program is rather fragile. When a local editor is started SiteBuilder creates a special file for it. This file is essentially a copy of the file existing in the user's edit area. This file can only be created by SiteBuilder, and it will be removed as soon as the version control functions are used.

If there are problems with saving a file, maybe because the version control functions have been used, there is a remedy. The file can always be saved on the local hard disk and then uploaded to SiteBuilder with the Upload function.

2.6.2 The built-in editor

The built-in, web-based editor has the advantage that it is always available, regardless of which computer or operating system is used. It is however lacking in usability, consisting of a text-edit window and nothing more. The web-based editor can only handle text and HTML files.

3 The SiteBuilder Editor

The SiteBuilder Editor is where all actions concerning editing and publishing of the site's web pages take place. It consists of four tabs: *Files*, *Access Control*, *Search Engine*, *Configuration* and *Docs*. The Files tab is for all file related actions, such as creating, editing, downloading and uploading files. The Access Control tab is for the access control system, and is only available to users with sufficient permissions. The Plugins tab holds extra products, such as LogView. It will only be shown if one or several of those products are installed and if the user has sufficient permissions to access them. The *Configuration* tab handles user customization as well as configuration of the SiteBuilder Editor. The Docs tab contains the available manuals and tutorials.

The SiteBuilder Editor is reached using a web browser. By default the SiteBuilder Editor is found on: `http://your_site/edit/`, but this can be customized at each site.

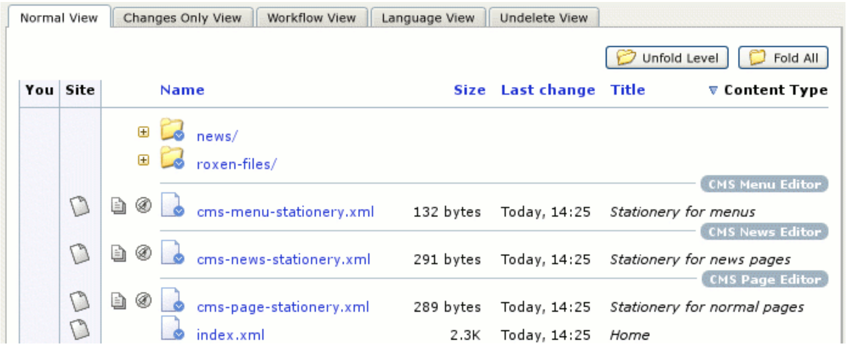
Usually the actual editing of files will be handled by client programs running on the user's computer. They are launched from within SiteBuilder by a helper program, *Roxen Application Launcher*, which must be installed on the user's computer. It is also possible to use FTP to access the files handled by SiteBuilder.

3.1 Getting Started

Most of the time spent in the SiteBuilder Editor will be under the Files tab. From here all modifications to the files on the site can be made. The page consists of two major parts; a column of buttons to the left and the file system to the right, see pictures below.

3.1.1 File System

SiteBuilder stores all files in a version controlled repository which does not correspond to a file system on your local computer. However, this does not make the file system much different from other file systems.



You	Site	Name	Size	Last change	Title	▿ Content Type
		⊞ news/				
		⊞ roxen-files/				
		cms-menu-stationery.xml	132 bytes	Today, 14:25	*Stationery for menus*	CMS Menu Editor
		cms-news-stationery.xml	291 bytes	Today, 14:25	*Stationery for news pages*	CMS News Editor
		cms-page-stationery.xml	289 bytes	Today, 14:25	*Stationery for normal pages*	CMS Page Editor
		index.xml	2.3K	Today, 14:25	*Home*	

The file system

3.1.2 File Types

The type of a file is stored in its metadata. The file type controls how it will be handled by SiteBuilder. A Word document will be handled differently than an image or an HTML file. The file type determines which editor to use when editing the file, how to upload and download the file as well as how the version control system will handle it. It is important that the file type is correct, otherwise it will be impossible for SiteBuilder to handle it correctly.

The file type is related to the file extension, i.e. the part of file name after the dot. SiteBuilder does not use the file extension internally, it uses the metadata instead. But when interacting with other systems the file extension will be used as a hint. When uploading a file with a .doc extension SiteBuilder will assume it is a Word file. In case the assumption is wrong the user has opportunity to correct it.

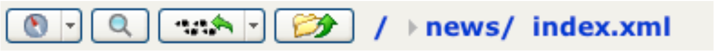
3.1.3 Focusing

Focusing on a file or directory is done by clicking on its name. Focusing on a file will show information about the file and allows for work on that file. Focusing on a directory shows a file listing of that directory and allows actions that can be performed on that directory. By clicking on the arrow to the left of a directory it is possible to view a directory listing of that directory without focusing on it.

Each file in the directory listings are listed with an icon representing its file type, one or two icons representing the version control status of the file, the file name, the size of the file and finally the file's title. The version control icons are explained further in the Version Control chapter. The title of the file is fetched from its metadata.

3.1.4 Navigation Buttons

You can browse the file system by clicking on the file name of each directory or file. This method is rather slow if you have a large file system with many files or many levels of subdirectories. Below the tabs you find some buttons that can help you browse the file system faster.

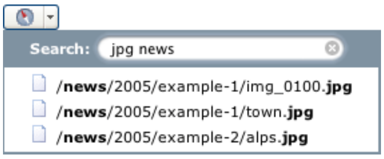


Navigation buttons

Compass icon

Shortcuts to each directory down to a few levels of subdirectories plus a quick search function for file and directory names.

To find file or directories, simply start typing your search string and wait for a second or two. You can include multiple search strings separated by space:



Search: jpg news

/**news**/2005/example-1/img_0100.**jpg**
/**news**/2005/example-1/town.**jpg**
/**news**/2005/example-2/alps.**jpg**

If too many files are found, only the first 30 items will be displayed.

By default the search will not include the /roxen-files/ directory structure. This behavior can be changed in the user preferences dialog by switching the "Enable site navigation popup" setting to "Yes, enabled".

Content search

Opens a popup window where you can search for text inside files. The search window contains settings to limit the search to e.g. particular path names or file types. This functionality is quite different compared to the standard search engine in that it’s fully aware of markup and special characters that a regular search indexer ignores.

History

Shortcuts to pages in the file system visited previously during the session.

Go Up

Focus the parent directory.

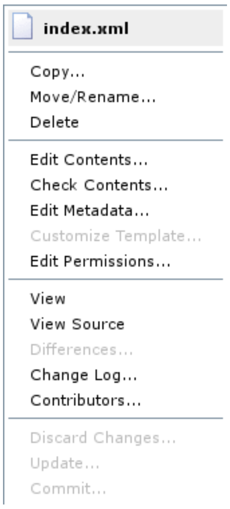
In addition to these tools, note that you can click on each segment of the path name for the current file or directory in order to jump directly to that location.

Note!

Some of the functions above may not be available if the site administrator has not added the Site News module to the site's configuration.

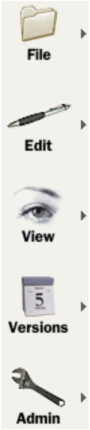
3.1.5 Action Menu

If you click on the icon of a file or directory you will get a pop-up menu with some of the actions available for the file. When this action menu is available it is indicated by a small triangle on the icon.



The pop-up action menu for an XML file.

3.1.6 Function Buttons



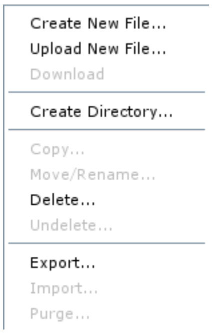
When clicking on the buttons in the left margin you activate pop-up menus with actions that can be performed on the focused file or directory. The actions are grouped according to their area of use. The available choices in these menus depend on the focused file or directory. They are also context sensitive; the user's access control permissions, the version control status as well as file type determines whether a choice will be available or not. For instance, if focusing on a directory where you are not allowed to change anything, most choices will be greyed out.

Note!

Actions ending with three dots indicate that a wizard page will be opened.

3.1.7 File

This menu contains actions changing the file system, such as adding or removing files or directories.



The File menu

Create New File...

Creates a new file.

Upload New File...

Creating a new file with the contents uploaded from the local file system.

Download

Download a copy of the file to the local file system, e.g. for off-line editing.

Create Directory...

Creates a new directory.

Copy...

Copies a file or directory to another location.

Move/Rename...

Renames a file or directory or moves it to another location.

Delete...

Deletes this file or directory. Files will only be deleted in the current user's edit area. Directories will be deleted together with all files and subdirectories.

Undelete...

Undeletes the file or directory.

Export...

Exports a section of the site to a self-contained archive on the server, and optionally downloads it to your local computer as a Zip file. This action is only visible to users with write permissions for the “Site Export/Import” protection class.

Import...

Imports a previously exported archive into the repository. This action requires the same permission as Export.

Purge...

Deletes the file or directory completely from the repository. This action is only visible to users with at least read permission for the “Purge file/directories” protection class.

Empty Trash...

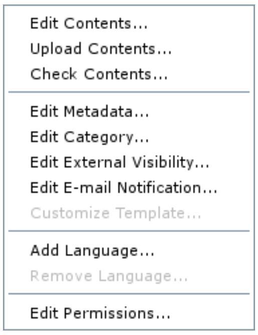
Performs a purge on files and directories that are deleted. This action requires the same permission as Purge.

Reassign Unique Identifier...

Displays and optionally resets the Unique Identifier for a file or directory. This identifier is a property used to e.g. track links between pages.

3.1.8 Edit

Here you find actions changing the content or status of a file.



The Edit menu

Edit Contents

Launches the chosen editor for that file type.

Edit as Alternate Type...

Sometimes there's a secondary content type defined for a file type. If the secondary type is XML, this option will show as "Edit as XML...". This will then launch the editor that would be launched for that secondary type. See the Administrator Manual for more info.

Upload Contents...

Upload a local copy of the file from the local file system.

Check Contents...

Checks the contents of a file for syntax errors. Only available for certain file types, e.g. XML-files.

Edit Metadata...

Editor for the metadata of the file, such as its file type, the current language and the template it will use.

Edit Category...

Lets you add or remove categories for a file.

Edit External Visibility...

Control if and when a file should be publicly accessible.

Edit E-mail Notification...

Configures e-mail reminders attached to events such as file commit, external visibility changes. A notification can also be associated to a specific date.

Customize Template...

Customize template parameters. Only available for XSL templates using certain parameter tags.

Add Language...

Add a language to the file. Some language-specific data will be added.

Remove Language...

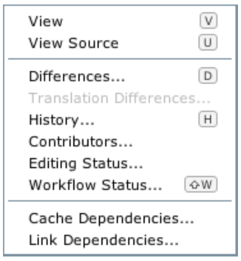
Remove an existing language and its specific data from the file.

Edit Permissions...

View and edit current directory or file permissions.

3.1.9 View

Options for viewing the content or status of the file.



The View menu

View

View the file in another browser window.

View source...

View the source of the file in another browser window.

Differences...

View the changes made to your local copy since you checked it out from the repository.

Translation differences...

View the changes in two different languages since both languages were last marked as "Up to date" simultaneously.

History...

Shows the version control log for the file or directory.

Contributors...

Shows the source of the file with annotations about when each line was changed and by whom.

Editing status...

Shows the status of all copies of the file, checked out by different users.

Workflow Status...

Shows details concerning a file that is currently, or has been, tracked in a workflow process.

Cache Dependencies...

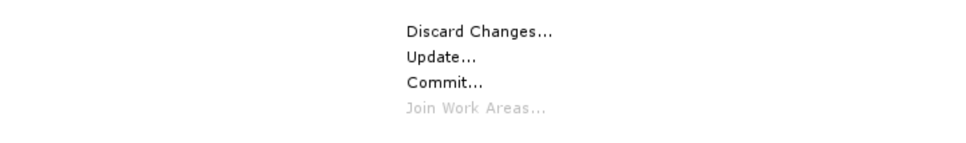
Presents information maintained by the persistent disk cache for this file.

Link Dependencies...

Shows data from the link manager module about links for this file.

3.1.10 Versions

This menu contains actions related to the version control system.



The Versions menu

Discard Changes...

Cancel the changes made to the file(s) since checked out to your edit area, i e remove the copy from your edit area.

Update...

Update the file(s) in the user's edit area with the latest version(s) from the repository.

Commit...

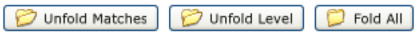
Commit the file or files from the users edit area to the repository.

Join Work Areas...

Join files in two different work areas.

3.1.11 View Tabs

When focusing on a directory, five mode tabs are shown above the file system area. The available function and *Fold/Unfold* buttons varies between the different modes.



The Changes Only tab's Fold and Unfold buttons

Normal View

The normal file view. Shows all directories and files available in the directory structure.

Changes Only View

Will only show files present in your edit area. Unfold matches will unfold all directories containing changed files.

Language View

Shows only multi language files, i.e. files with at least one language added. If the metadata entry Translation status is not marked as Up-to-date the language icon is shown as a half flag. Unfold matches will unfold all directories containing multi language files.

Undelete View

Enters the undelete mode where removed files can be undeleted. Undelete mode is represented by a black and yellow bar above and below the file system area and only a few of the function buttons will be available, depending on which file is focused on.

3.2 Creating Files/Directories

To create a new file or directory in the directory you are, choose *Create new file*, *Upload new file* or *Create directory* from the *File* menu to start the wizard that guides you through the creation process.

Before creating a new file or directory it is important to be aware of the fact that it is not possible to give the same name on files and directories in the same directory. If a name has been given to a file there can never exist a directory with the same name, even though the file has been removed, and vice versa. Thus if a directory index.html/ is created no file named index.html can be created in that directory.

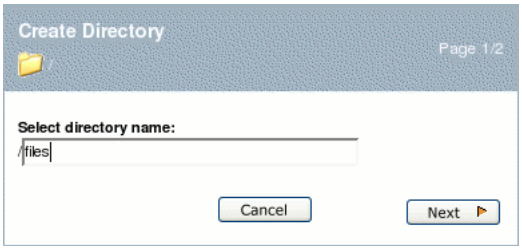
Note!

There is a Purge command that an administrator can use to permanently delete a file or directory so that its name can be reused. However, this command removes the full version history and cannot be undone.

3.2.1 Create Directory

A directory is created by first focusing on the directory in which the new directory should be created, then select Create directory from the File menu. A wizard will guide you through the creation process.

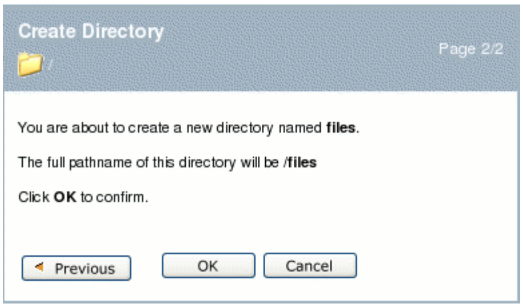
In the first part of the wizard the name of the new directory is chosen.



Select directory name:
/files

Choose directory name

On the second page of the wizard the choice is confirmed. When the Ok button is pressed the new directory will be created.



You are about to create a new directory named **files**.

The full pathname of this directory will be /**files**

Click **OK** to confirm.

Confirm directory path

3.2.2 Create New Files

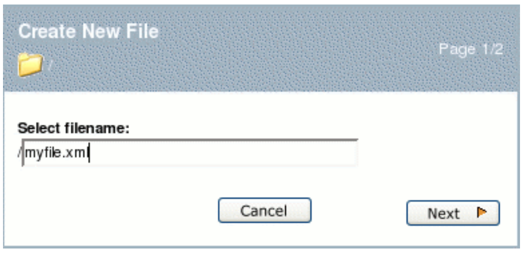
To create a new file select either Create new file from the File menu to create a new file, or select Upload new file to upload an existing file from the local file system. More information about uploading files can be found in the Uploading/Downloading Files page.

Note!

> When creating a new file it is necessary to focus on the correct directory where the new file should be stored.

When selecting Create new file from the menu a wizard is launched that guides you through the creation process.

The first page in the wizard is used to name the new file. It is not possible to create a file with the same name as an existing file or directory.

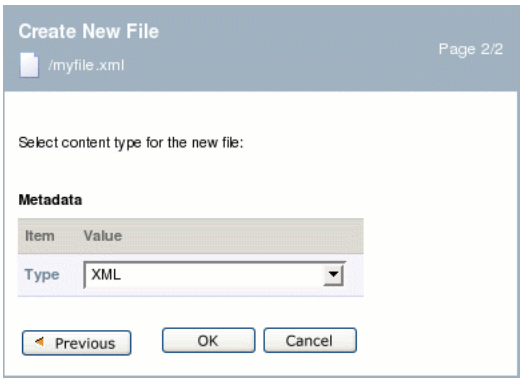


Select filename:
/myfile.xml

Select file name

The second page of the wizard is used to confirm the content type of the file. It is especially important to enter the correct type for the file, as the file type determines how SiteBuilder will handle the file. For HTML files the template is also important, as it determines what layout will be added to the content.

SiteBuilder will recognize most file types from their extensions, so by naming, for example, HTML documents like name.html the correct type will be chosen automatically. SiteBuilder can also be configured to recognize additional file types of your choice as explained in the administrator's manual.



Select content type for the new file:

Metadata

Item	Value
Type	XML

Select content type

To edit the new file, select Edit contents from the Edit menu.

3.3 Copying/Moving Files

Copying and moving files and directories is easily done in SiteBuilder. A wizard guides you through each process. The wizard is similar for both operations and easy to use.

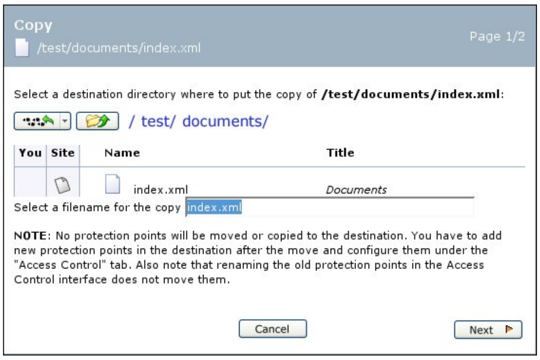
3.3.1 Copy

The first step is to focus on the file or directory that is to be copied. Then select Copy from the File menu to launch the wizard that will guide you through the process.

The first page of the wizard is used to find the destination directory where the copy should be saved, and possibly change the name of the copy. In the example the file /test/documents/index.html is copied to /test/user/document.html.

Let us start by viewing the /test/documents/ directory. The file system within the wizard is navigated in the same way as the regular SiteBuilder file system.

Choose the Go up button to navigate upwards in the file system towards the destination directory.



Copy

/test/documents/index.xml

Page 1/2

Select a destination directory where to put the copy of **/test/documents/index.xml**:

/ test/ documents/

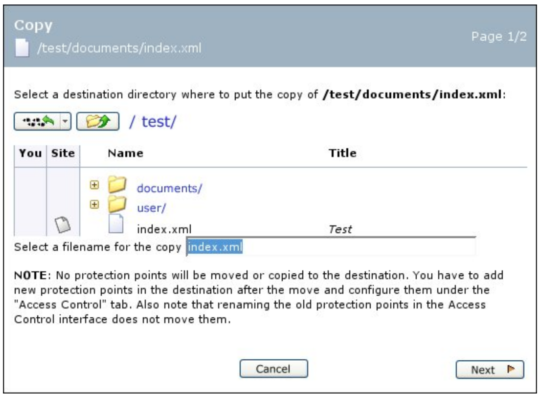
You	Site	Name	Title
		index.xml	*Documents*

Select a filename for the copy index.xml

NOTE: No protection points will be moved or copied to the destination. You have to add new protection points in the destination after the move and configure them under the "Access Control" tab. Also note that renaming the old protection points in the Access Control interface does not move them.

/test/documents/ directory

After having followed the Go up link you will be looking at the /test/ directory. From here you can choose the user/ directory.



Copy

/test/documents/index.xml

Page 1/2

Select a destination directory where to put the copy of **/test/documents/index.xml**:

/ test/

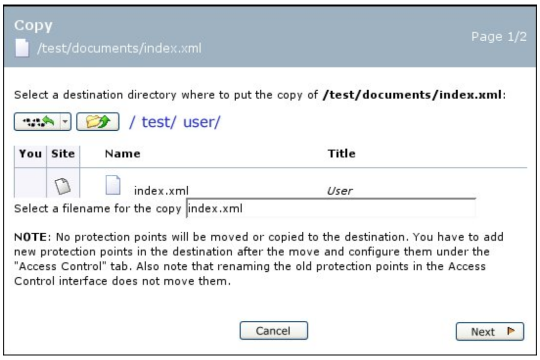
You	Site	Name	Title
		documents/	
		user/	
		index.xml	*Test*

Select a filename for the copy index.xml

NOTE: No protection points will be moved or copied to the destination. You have to add new protection points in the destination after the move and configure them under the "Access Control" tab. Also note that renaming the old protection points in the Access Control interface does not move them.

/test/ directory

Now the chosen destination directory, /test/user/, is shown. The file name is changed to document.html before following the next step in the wizard.



Copy

/test/documents/index.xml

Page 1/2

Select a destination directory where to put the copy of **/test/documents/index.xml**:

/ test/ user/

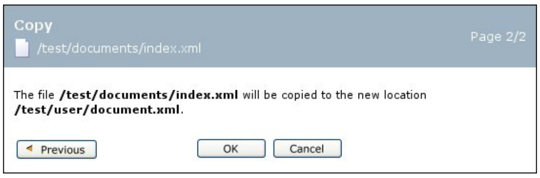
You	Site	Name	Title
		index.xml	*User*

Select a filename for the copy index.xml

NOTE: No protection points will be moved or copied to the destination. You have to add new protection points in the destination after the move and configure them under the "Access Control" tab. Also note that renaming the old protection points in the Access Control interface does not move them.

/test/user/ directory

The second page of the wizard asks for confirmation of the operation. When pressing Ok the copying will take place. It will however be necessary to commit the changed file(s) before other users can see the result of the operation.



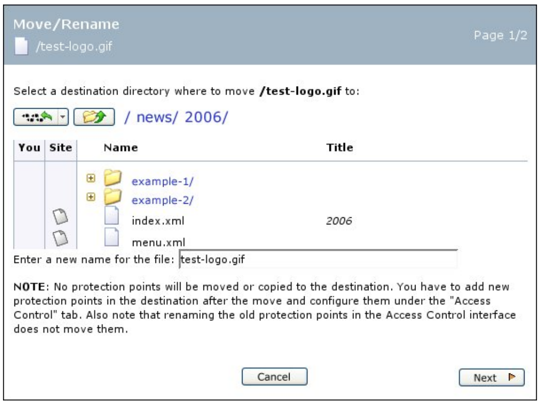
The file **/test/documents/index.xml** will be copied to the new location **/test/user/document.xml**.

Confirmation of the copy operation

In case of copying a directory, the second page of the wizard will also ask for a log message. As with all directory operations it will be committed to the repository immediately, thus the need for a log message. All files and subdirectories within the directory will be copied as well, keeping their original names.

3.3.2 Move/Rename

Moving or renaming a file or directory follows the same procedure as copy. Focus on the file or directory that should be moved and then select Move/Rename from the File menu. Follow the instructions in the wizard and - when the wizard is finished - commit the changes.



Select a destination directory where to move **/test-logo.gif** to:

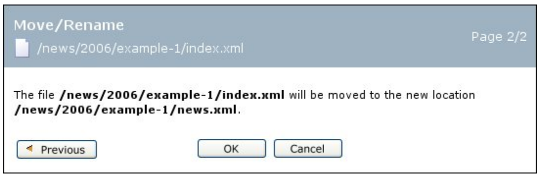
You	Site	Name	Title
		example-1/	
		example-2/	
		index.xml	*2006*
		menu.xml	

Enter a new name for the file: test-logo.gif

NOTE: No protection points will be moved or copied to the destination. You have to add new protection points in the destination after the move and configure them under the "Access Control" tab. Also note that renaming the old protection points in the Access Control interface does not move them.

Move/Rename wizard

As with the copy wizard it is necessary to confirm the move.



The file **/news/2006/example-1/index.xml** will be moved to the new location **/news/2006/example-1/news.xml**.

Confirmation of the move operation

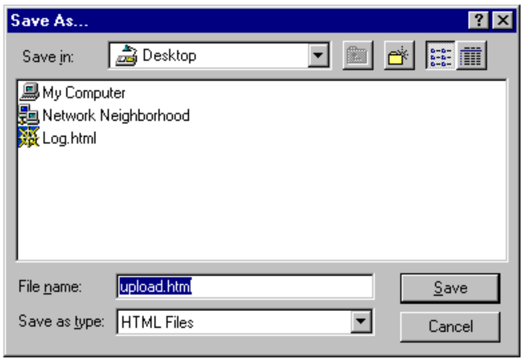
In case of moving a directory the second page of the wizard will also ask for a log message. As with all directory operations it will be committed to the repository immediately, thus the need for a log message. All files and subdirectories within the directory will be moved as well.

3.4 Uploading/Downloading Files

Files are copied between the local file system and SiteBuilder's file system by the Download and Upload functions, or by using ftp.

3.4.1 Download

A file is downloaded by focusing on the file and clicking on the Download button in the File menu. The file will be copied using the save file function of the browser, which allows for browsing the local file system for a directory in which to save the file.



Save in: Desktop

File name: upload.html

Save as type: HTML Files

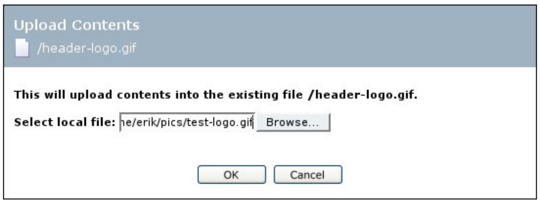
An example download

When a file is downloaded it will be copied to your edit area, so that the version control system can keep track of the changes made to the file. This is important if someone else is committing changes to the file while you are editing the content.

The setting Text download convention in the User preferences (Configuration tab) controls what line breaks will be used when downloading a text or HTML file. It is important to set this since different operating systems handle line breaks differently.

3.4.2 Upload

To upload a file's contents, focus on the file and click on the Upload contents button in the Edit menu to browse the local file system for the right file to be uploaded.



This will upload contents into the existing file /header-logo.gif.

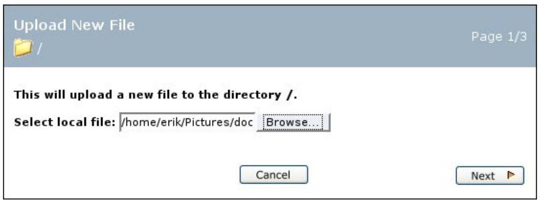
Select local file: he/erik/pics/test-logo.gif

Upload wizard

Note!

> It is important that you focus on the correct file when uploading contents. If you focus on the wrong file you will overwrite that file.

To upload a new file a directory must be focused, not a file. By selecting Upload new file from the File menu a wizard will be activated. The wizard allows for browsing the local file system in search for the correct file.



Upload New File

/

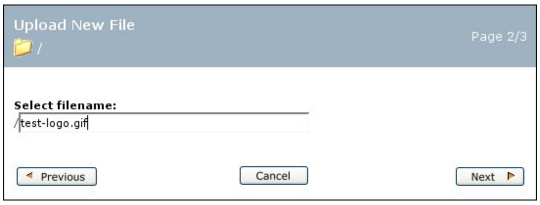
Page 1/3

This will upload a new file to the directory /.

Select local file: /home/erik/Pictures/doc

Upload new file wizard

The second page of the wizard provides the possibility of giving the file a more appropriate name than before. As it also gives the directory path to the new file it also acts as a reminder to minimize the possibility of misplacing the file.



Upload New File

/

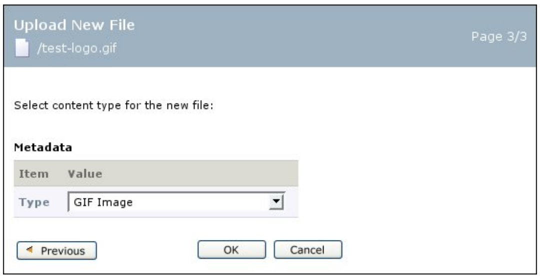
Page 2/3

Select filename:

/test-logo.gif

Select filename

The third page of the wizard is for selecting the content type. If the file was named with a proper extension, in this case ".gif", the correct content type will be set automatically for file types known by the server.



Upload New File

/test-logo.gif

Page 3/3

Select content type for the new file:

Metadata

Item	Value
Type	GIF Image

Select content type

When uploading a text or HTML file it is not necessary to have the right Text download convention in the User preferences (Configuration tab) set, since SiteBuilder will handle all possible line breaks when uploading.

3.4.3 FTP

It is possible to use ftp to upload and download files to SiteBuilder's file system. If, and where, the ftp port can be found is installation dependent. Ask your administrator for details.

To use a web browser as a ftp client you type a ftp URL, usually ftp://user name@your site/, in the location toolbar. When using a special ftp client you type in the user name, password and the name of your site separately.

As when downloading through the Download button, files you download through ftp will be stored in your edit area. The Text download convention in the User preferences (Configuration tab) will be used to convert the line breaks as well.

Note!

It is not possible to perform all actions, that are normally available within the SiteBuilder Editor, through ftp (e.g. committing files).

3.5 Deleting Files

3.5.1 Files

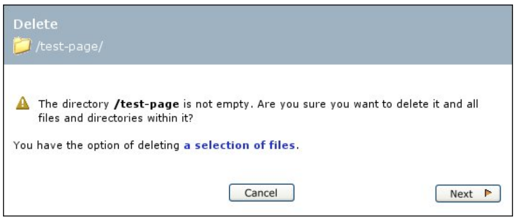
A file is deleted by focusing on it and selecting Delete from the File menu. To finalize the deletion of a file from the file system the file must be committed. However, when a file has been deleted it can be undeleted by entering Undelete View which is described in the Version Control chapter.

3.5.2 Multiple files at once

Deleting multiple files at once can be done by selecting Delete from the File menu for the directory in which the files are located. This will bring up the screens illustrated in the next section. In the first screen, select the link titled deleting a selection of files to get a list of files where you can check or uncheck several files at once.

3.5.3 Directories

A directory is deleted by focusing on it and selecting Delete from the File menu. All files and subdirectories within the directory will be deleted as well. As with all directory operations it will be committed to the repository immediately. It will of course be possible to undelete the directory and any files and subdirectories by entering the Undelete View.



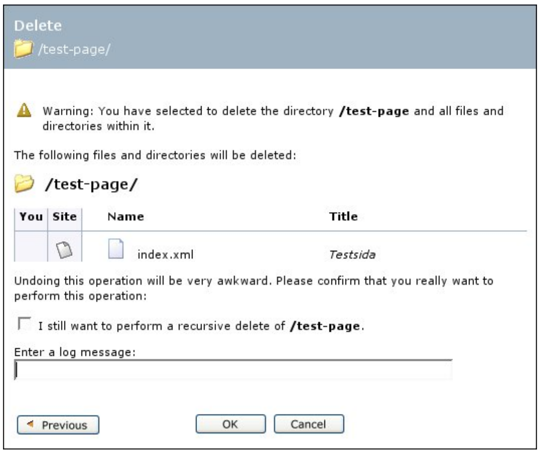
Delete
/test-page/

The directory **/test-page** is not empty. Are you sure you want to delete it and all files and directories within it?

You have the option of deleting **a selection of files**.

The delete directory wizard

If the directory is not empty, the wizard will ask to confirm if the deletion is to take place. To avoid deleting directories by mistake, you must confirm by writing a log message and selecting the check-box if the deletion is to take place.



Delete

/test-page/

Warning: You have selected to delete the directory **/test-page** and all files and directories within it.

The following files and directories will be deleted:

/test-page/

You	Site	Name	Title
		index.xml	*Testsida*

Undoing this operation will be very awkward. Please confirm that you really want to perform this operation:

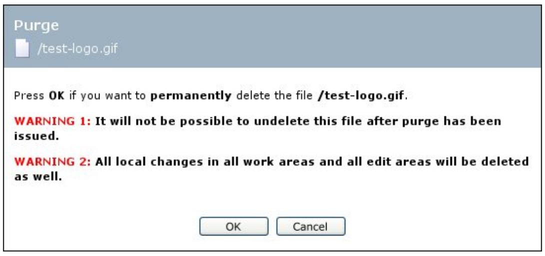
□ I still want to perform a recursive delete of **/test-page**.

Enter a log message:

Confirmation page

3.5.4 Purge

If you have write permission for the Purge protection class you will be able to remove files or directories completely from the repository and all edit areas. This means they cannot be restored in Undelete View. Purge is found in the File menu (but will only be visible there if you have at least read permission for the protection class). The wizard will ask for confirmation before deleting any files.



Purge

/test-logo.gif

Press **OK** if you want to **permanently** delete the file **/test-logo.gif**.

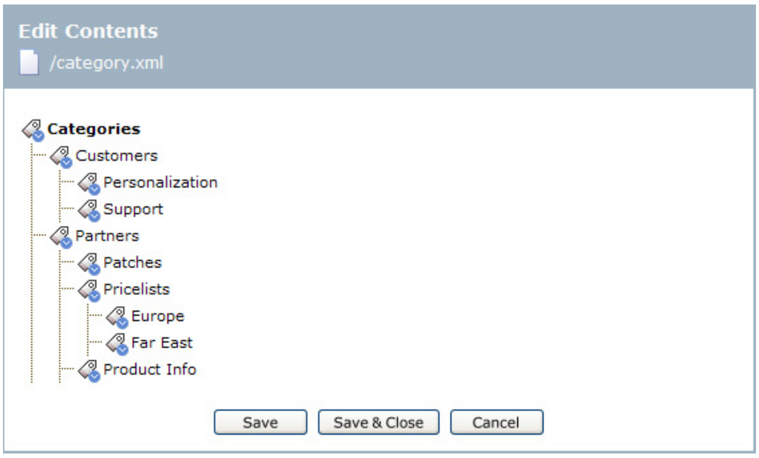
WARNING 1: It will not be possible to undelete this file after purge has been issued.

WARNING 2: All local changes in all work areas and all edit areas will be deleted as well.

The Purge warning page

3.6 Creating a Category Tree

A category tree is represented by a file with the type CMS Category Tree. This is a regular XML file which can be edited easily using the graphical category tree editor, see screen shot below:



Editing a category tree.

Category tree nodes can be added, renamed, deleted and be made related using the pop-up menu found by clicking on a category node icon.

A file can be assigned to any number of categories in any number of category trees. All category tree files which are found by *Smart Search* in the current directory for a file or any of its parent directories, and have read permissions, can be used for the file.

3.7 Edit Permissions

3.7.1 Inherited versus explicit permissions

Access permissions control which users that can see or edit a file or directory. Roxen CMS maintains a list of "protection points" internally and lets you edit the settings in the Access Control tab. However, as a shortcut you can open the Edit Permissions... wizard in the Access menu for any file or directory from the Files tab. Permissions are inherited by subdirectories unless they are given other settings explicitly.

The inheritance concept is not limited to a single directory level. It can be applied to any number of directories and their subdirectories. Initially the whole site will be controlled by the access permissions on the root directory, in addition to permissions defined for each work area (which have higher priority than any file/directory protection points).

At the right-hand side of the Edit Permissions wizard screen you will find a button with one of the following titles (depending on the current page status):

Set explicit permissions...

> Asks you to verify that you want to create a protection point for the current file/directory. Click "Create" to proceed. The newly created protection point will get its initial settings from the closest parent directory for which the access permissions have been customized.

Remove explicit permissions...

> Removes any custom access permissions for this page/directory. Click "Remove" to confirm that this is what you want. The new settings will be inherited from the closest parent directory for which the access permissions have been customized.

3.7.2 Assigning permissions to groups

Permissions are always given to groups of users. If you want to give individual users specific permissions they need to be put into separate groups first (which is a task for the administrator).

If the file/directory is using inherited permissions, click the "Set explicit permissions..." button as described above. The "Permissions" table on the left-hand side describing the current permissions will now offer a number of buttons where you can add or remove groups as well as changing their settings.

By clicking "Add group" you get a search box where you can find and select one or more groups to add to the permission table. Existing groups can be removed using the red icon in the "Remove" column. Use the popup menu in the "Perm" column to set the permission to one of "None", "Read" or "Write".

3.7.3 Indirect membership

In some situations you may see a second table in the wizard. This table is titled "Permissions inherited by group memberships" and lists those groups which have been given permissions through membership in one of the groups you've included in the first table. To override the permissions caused by inherited membership you must add the group to the "Permissions" table.

4 Editing Files

A file in SiteBuilder consists of the file contents and its associated metadata (see the Metadata chapter). The metadata contains information about the file type, the title of the page, what template should be used as well as information about the page sent to search engines.

Each HTML file is combined with a template before being sent to the user. Usually layout and navigation support are handled entirely by the template, freeing the writer of the content file to concentrate fully on the actual content. It is however still possible to use HTML or RXML tags in the content file to emphasize text, create diagrams and so forth. Furthermore, the template may provide you with additional tags for formatting, automatic index generation and so on.

Files are usually edited on-line while the user are connected to the SiteBuilder Editor. This is the easiest way, and all functionality of SiteBuilder will be available. It is however also possible to edit files off-line, by first downloading and the uploading the files.

On-line editing is either done through the built-in web-based editor, a local program started by the Roxen Application Launcher or a local program capable of accessing SiteBuilder through ftp. To use a local program is often best since the user then can use the HTML editor, drawing program or word processor she is familiar with.

4.1 The Built-in Editor

The built-in web-based editor consists of a simple form that lets the user edit text files. Text files include files configured to use the text handler and also HTML, XML and XSL files.

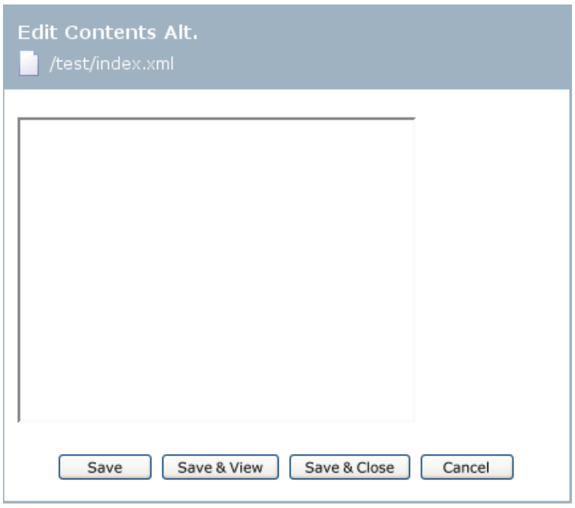
To edit a file, focus on the file and choose Edit contents from the Edit menu. When having finished editing, select one of the buttons at the bottom. It is possible either to throw the changes away by clicking on Cancel or selecting Ok to save the changes to the user's edit area. For some file types you have the two following choices instead of Ok to save the changes:

Save

Saves the changes and continue editing.

Save & Close

Saves the changes and closes the edit form.



The built-in editor

It is possible to alter the width and height of the edit window by selecting the User preferences button on the Configuration tab page (Edit window rows and Edit window columns settings). It is a good idea to change the size of the edit window to better suit the size of the screen, which in turn makes it easier to work.

When saving changes to a file, the changes are not automatically transferred to the live site but are only saved to the current user's edit area. It is always necessary to use the Commit button, otherwise no one else will be able to see them.

Due to restrictions regarding forms in some browsers, it is not always possible to edit large files (around 1 MB or more).

4.2 Roxen Application Launcher

Users using SiteBuilder extensively will most likely use a programs running on their computer to edit files, such as HTML editors, drawing programs or word processors. Roxen Application Launcher lets the user do exactly that. When the user presses the Edit button in the SiteBuilder Editor the appropriate program will be started with the correct file loaded.

You can read more about Roxen Application Launcher in a later section of this manual.

4.3 Check Contents

When focusing XML files or XSL templates you can be warned of syntax errors. You can perform a syntax check of the contents by selecting the Check contents button in the Edit menu. A wizard will show you the errors found. Some errors can be corrected automatically and are shown in blue while errors the check wizard cannot correct are shown in red. Select the check-box and press OK to correct the recoverable errors.

It is possible that the check wizard will not always produce a syntactically correct file when correcting errors, but it usually helps you find the errors easier.

4.4 FTP

Files in SiteBuilder can be accessed by FTP if the server has an FTP port configured that is connected to the SiteBuilder. How to configure the FTP port is described in the Installation section of the Administrator's manual.

When using FTP, SiteBuilder's file structure appears as a top-level directory containing one sub directory for each Workarea. Inside each Work area directory, the rest of the path is identical to the part that is displayed in the ordinary SiteBuilder interface. In other words, the FTP path is:

ftp://hostname:port/workarea/documentpath

The FTP interface does not support access to metadata. When creating new files through FTP, SiteBuilder will make an educated guess as to what kind of file it is, based on its extension, such as .html for HTML files, etc. HTML files will have their title extracted, but not removed, and inserted into the metadata, if SiteBuilder can find a header section containing a title block in the new file.

No version control functions are available through FTP so to commit the file the user has to use the SiteBuilder Editor. This means that after the user has saved a file and chosen "Update", she has to close the file and reload it to get the latest version.

4.5 Editing in Microsoft Word

The Roxen Application Launcher makes it easy to edit files using Microsoft Word. Application Launcher should be configured by the administrator so that when a user focuses on a text or HTML file, she will only have to press Edit and the file will be opened in Microsoft Word.

See the Application Launcher section for more information.

4.6 Emacs

Lots of people prefer Emacs to many HTML editors due to its ease of use when familiar with it. For those who never or seldom use Emacs it can be very annoying and difficult to use.

4.6.1 Ange-ftp or efs

Ange-ftp is an ftp program that makes it possible to fetch files from an ftp site, alter them in Emacs and then upload them again. All is done by using standard Emacs commands. The URL ange-ftp uses is slightly different from what we are used to from web browsers. The syntax when logging on to an ftp site using ange-ftp is:

C-x C-f (Control-x Control-f) /username@URL.of.your.site C-q (Control-q, spacebar) portnr:/ Tab

It would look like this in the minibuffer window (bottom of the Emacs window): /username@URL.of.your.site 21:/.

If no port number is needed, disregard from the part: C-q (Control-q, spacebar) portnr:/.

When hitting Return, Emacs will ask for a password. When the password has been given and Return has been pressed Emacs will present the sites directory listing in the window.

If the user would press the Tab button instead, Emacs would first ask for the password and then give a tab-completion if the Tab button is pressed again. If there are multiple choices, Emacs will list them at the bottom of the window.

After logging in with ange-ftp it is possible to edit the files as one normally edits a file in Emacs.

4.6.2 Create Files and Directories

A file is created by pressing C-x C-f and selecting a file name followed by Return.

A directory is created by pressing C-x C-f and typing directoryname/filename followed by Return. Emacs then asks the user to perform the operation Create directory by pressing M-x dired-create-directory and pressing Return twice. Usually it is not necessary to create a file at the same time as creating a directory when using ange-ftp. However, for others to see the directory in SiteBuilder it is necessary that it contains at least one file.

4.6.3 Save and Commit

After the editing is done, the file is saved in Emacs by pressing C-x C-s. This operation saves the file in the local edit area by uploading the file via ftp. To make it possible for others to see the file it must be committed.

Note!

> After saving a file with ange-ftp, you must either close it or reload it from the site before continuing to edit its contents, or you will risk undetected conflicts with other people editing the file on the server.

4.6.4 Emacs and SiteBuilder

Does ange-ftp sound too difficult? In SiteBuilder the administrator can change the file type settings to make Emacs open a file automatically when pressing the Edit contents button. This feature makes it possible to take advantage of Emacs' powerful editing tools without needing to think about making the ftp connection.

However, this does not make ange-ftp unnecessary because the connection Emacs + SiteBuilder only works for one file at a time and ange-ftp is still a powerful tool for handling several files. This means that the user must create files and directories inside SiteBuilder and for each file she wishes to edit, a new Emacs must be started. Also, when the user has saved her file once it is not possible to save it again unless she presses the Edit contents button.

5 Metadata

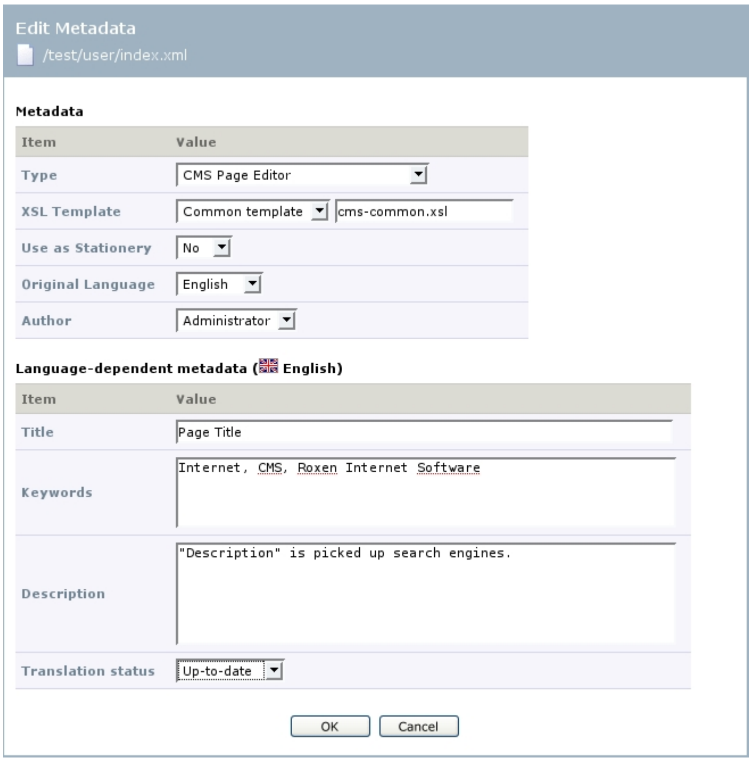
Metadata is information that tells the server how to handle a particular file. It can also help search engines to index the file correctly.

There is a special editor for setting the correct metadata of a file. Metadata can also be set automatically when creating the file if there is a stationery for that particular file type.

A stationery is a "content template" for this kind of file. It contains a standard protocol with all the usual headlines and sections. The user's work is simply to fill it out with her specific protocol text.

To access the metadata editor, focus on the file and select Edit metadata.

In the editor the following metadata can be set:



Edit Metadata

/test/user/index.xml

Metadata

Item	Value
Type	CMS Page Editor
XSL Template	Common template cms-common.xsl
Use as Stationery	No
Original Language	English
Author	Administrator

Language-dependent metadata (English)

Item	Value
Title	Page Title
Keywords	Internet, CMS, Roxen Internet Software
Description	"Description" is picked up search engines.
Translation status	Up-to-date

The metadata editor

Type

The file type of the file. SiteBuilder treats each file according to its type. Thus it is very important to set the correct type. Among other things the file type controls which editor will be used to edit the file and how the version control system will treat the file. Which file types are available can be configured at each site.

Unlike other metadata fields there is a Change button after the type select box. To change the type of a file it might be necessary to press the Change button (on some systems, the editor will be reloaded automatically). This is because the type controls which other metadata fields will be available. The template field will only be available for XML and HTML files.

Auto Output Conversion

You can let Roxen CMS automatically convert images from one format to another when the image is requested by a web browser. This can be used to generate, for example, a GIF version of an Adobe Photoshop file. The file stored in the repository will not be affected and Roxen CMS will keep the generated image up-to-date with any changes to the original file.

Note that this metadata setting is only applicable to certain image types.

Template

For HTML files it becomes possible to choose a template. The template provides the content from the file with layout. What templates are available depends on what templates are available on the site.

XSL Template

For XML files it becomes possible to choose an XSL template. The templates available for selection are all XSL template files in the current directory and any of its parent directories, except those explicitly marked as not selectable.

Selectable

Only available for template files, deciding whether or not they should be selectable from the template selection menu.

Use as Stationery

If set to Yes, metadata settings can be copied automatically for all new files created in this directory. Select the metadata fields you want to copy using the check-boxes.

Original Language

Only available for multi-language files. This setting governs what language translations of the file should be based on. By default, it is set to the first language added to the file.

Author

Username of the person that created the file. Default is No author. This is only a description and does not affect the behavior of the file.

5.1 Language-dependent Metadata

Title

A title that describes the page more accurately than the file name. The title may be used as the page header from your template files, and will show up in the SiteBuilder Editor interface. By setting the title in file metadata instead of in a HTML tag <title>, it is easier for template authors to use the page title in navigation interfaces and similar.

Keywords

Document keywords that help search engines index the site better. Use of keywords can give a page better ranking in the search result, for instance.

Description

A document description that will be used in the search result given by search engines. If nothing is entered here, most search engines will only use a part of the text on the top of the page instead.

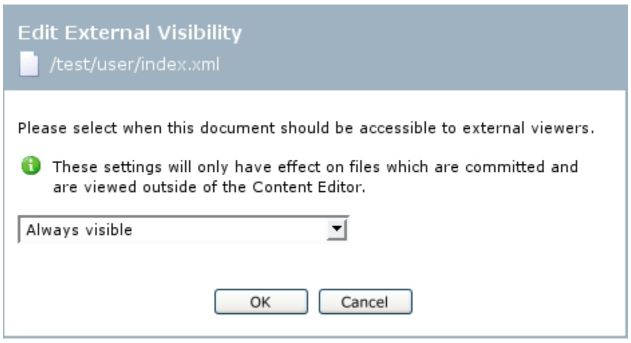
The description field can also be used in a navigation or site map interface.

Translation status

The translation statuses Up-to-date or In progress can be set from here.

5.2 External Visibility

Sometimes you might want to show a certain web page during a limited period of time. This can be configured in the Edit menu, Edit external visibility view.



Edit External Visibility

/test/user/index.xml

Please select when this document should be accessible to external viewers.

These settings will only have effect on files which are committed and are viewed outside of the Content Editor.

Always visible

The edit external visibility wizard

Always visible

This is default. The page will always be visible when browsing the site. Access to the page can of course still be limited by the access control system.edit/

Never visible

The page can only be viewed from inside the SiteBuilder Editor. No links will be shown to the page.

Visible until a specified time

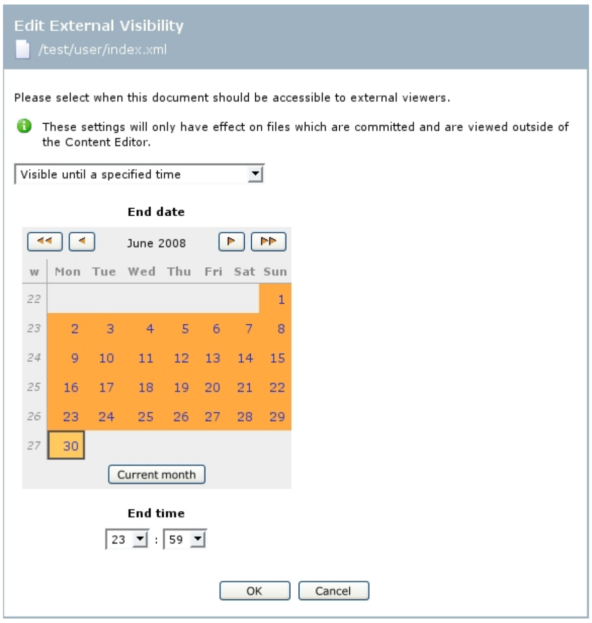
The page will only be visible until the specified time.

Visible after a specified time

The page will only be visible after the specified time.

Visible during a specified time interval

The page will only be visible during the specified time interval.


Edit External Visibility

/test/user/index.xml

Please select when this document should be accessible to external viewers.

These settings will only have effect on files which are committed and are viewed outside of the Content Editor.

Visible until a specified time

End date

June 2008

w	Mon	Tue	Wed	Thu	Fri	Sat	Sun
22							1
23	2	3	4	5	6	7	8
24	9	10	11	12	13	14	15
25	16	17	18	19	20	21	22
26	23	24	25	26	27	28	29
27	30						

End time

23 : 59

Selecting time interval

As shown in the picture above, the visibility time can be specified in 15 minute intervals. You can change month by clicking on one of the single arrow buttons or change year by clicking on one of the double arrow buttons. Select the date by clicking on it. The time period when the page will be visible will be shown with another background color. In the example above, the page will be visible until 23:59 June 30, 2008. Then the page will only be viewable from inside the SiteBuilder Editor.

5.3 E-mail Notification

5.3.1 Configuring e-mail notification for expired or changed pages

You can ask Roxen CMS to remind you of files which, for example, are about to expire or which have content that you want to update periodically. This notification will be sent as an e-mail to one ore more addresses of your choice.

Start by selecting "Edit e-mail notification" in the Edit menu. This will display a wizard where you can configure different parts of the notification:

Time of notification

The time may be either never, relative to another event, or an absolute date. The first and last choices are self-explanatory, while the middle one offers several variations; the event upon which the actual notification time will be calculated from can be any of the following:

- File publishing date
- Start of external visibility
- End of external visibility

Given an event the actual notification time can be offset a specified number of days. In case of visibility starting/ending dates the offset may be set to indicate an earlier date. (This is of course not applicable to publish time since that date isn't known beforehand.)

The event-relative times are dynamic and an outstanding notification will be rescheduled if the event it depends on is set to a different date. For example, suppose there is a notification registered for three weeks after the file publish date. Now, if the file is updated one week after the first publish date, the original notification will be canceled and replaced by another set to three weeks after the most recent publish date.

Check the "Repeat notification once a week" checkbox to have an event repeated weekly after the first notification time.

Recipients

The recipients of the notification message can be one or more e-mail addresses. If the file author's e-mail address is known to the Access Control system (e.g. when an LDAP server provides it) you can check the first checkbox to have it handled automatically. When the second checkbox is enabled you can enter a comma-separated list of addresses in the corresponding text field.

Message

By default the e-mail message will only contain the location of the file. If you enter a custom message it will be included in the e-mail as well.

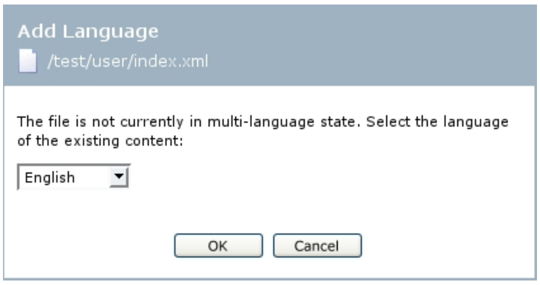
6 Multilingual Support

Sometimes you want your web pages to be available in more than one language. In SiteBuilder one or more languages can be chosen both for the content and the language-dependent metadata of a file.

When viewing the file, one language version will be shown at a time. Which language will be shown depends on how the file is viewed or on the settings of your browser. When using the View button in the SiteBuilder Editor the currently selected language will be shown. When viewing the site, the languages will be chosen according to the preferred languages in the settings of your browser.

6.1 Add Language

To add another language to a file you click on the Edit button and choose Add language while focusing on the file. Choose a language from the select box and confirm your choice by clicking on Ok.



Add Language

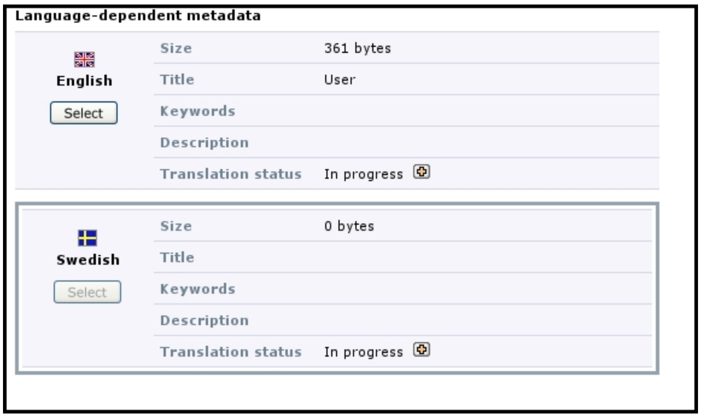
/test/user/index.xml

The file is not currently in multi-language state. Select the language of the existing content:

English

Add language wizard

By adding another language you get another set of language-dependent metadata. When you want to edit the metadata for a certain language you must first select the language by pressing the Select button representing the metadata of that language. Hereby you also select the default language for all language-dependent actions that can be done to the file in the SiteBuilder Editor.



Language-dependent metadata

English	Size	361 bytes
	Title	User
	Keywords	
	Description	
	Translation status	In progress
Swedish	Size	0 bytes
	Title	
	Keywords	
	Description	
	Translation status	In progress

Selected language

The language-dependent actions are marked with a flag in the Edit drop down menu. The action will be performed for the selected language by default. For some actions, such as choosing View source from the View menu, it is possible to choose another language from a select box.

6.2 Translation Differences

When working with files in two different languages you might want to compare the changes made in both languages. The Translation differences-wizard (found in the View menu) shows the changes made to both files since they were last marked as Up to date in the metadata entries.

7 Templates

This part of the manual is directed to the writers of the content and they should not need to bother too much about the overall layout of the site. A template provides the site with a backbone and a layout in which content can be entered easily without changing or disrupting the site layout.

The purpose of SiteBuilder is to free the webmaster and his co-workers from the tedious work of keeping a web site up to date and the layout consistent and instead letting people do what they do best. The template designers should concentrate on the site layout and the writers should keep the content up to date, not something in between.

A template can be made to give different degrees of control over the layout of a site to the writer.

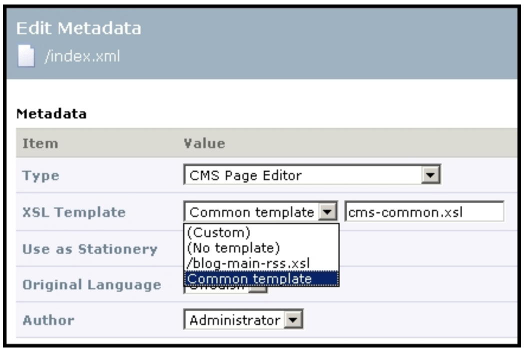
- Making the template in total control over the pages, putting the content into well confined areas which the writer cannot affect. This leaves little freedom for the writer, but instead requires very little knowledge of HTML or RXML.
- The template can be constructed for only controlling the most basic layout, like side bars, navigation etc. to be put on all pages of the site to give it a consistent look. By giving the writer a set of tags, she can give the page a personal touch.
- The template can also be made to provide the writer with a set of tags, letting him or her do the complete layout.

7.1 Choosing Template

Sometimes a web designer wants to have different layout on her site. A special celebration, event, time of year or simply new functionality might demand a different layout. With templates, this is easily achieved.

7.1.1 Changing Template

Changing template for a file is done by focusing on the file and then clicking on the Edit metadata button in the Edit menu. Choose the proper template from the template select box, which contains a list of all available templates. Confirm your choice by clicking on Ok at the bottom of the page.



Edit Metadata
/index.xml

Metadata

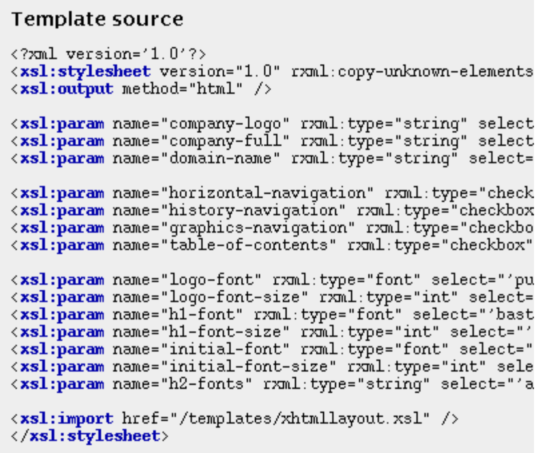
Item	Value
Type	CMS Page Editor
XSL Template	Common template cms-common.xsl
Use as Stationery	
Original Language	
Author	Administrator

Choosing template

7.2 Viewing the Template

The template (.xls) can be viewed like any other file. The point of viewing a template is, of course, to learn of the special tags created in it.

When focusing on a template file, a list of all tags defined in the template, and their arguments, will be shown below the metadata table. To view the source of the previewed tags, select View source from the View menu.



Template source

```
<?xml version='1.0'?>
<xsl:stylesheet version="1.0" rxml:copy-unknown-elements
<xsl:output method="html" />

<xsl:param name="company-logo" rxml:type="string" select
<xsl:param name="company-full" rxml:type="string" select
<xsl:param name="domain-name" rxml:type="string" select=

<xsl:param name="horizontal-navigation" rxml:type="check
<xsl:param name="history-navigation" rxml:type="checkbox
<xsl:param name="graphics-navigation" rxml:type="checkbo
<xsl:param name="table-of-contents" rxml:type="checkbox"

<xsl:param name="logo-font" rxml:type="font" select="'pu
<xsl:param name="logo-font-size" rxml:type="int" select=
<xsl:param name="h1-font" rxml:type="font" select="'bast
<xsl:param name="h1-font-size" rxml:type="int" select="'
<xsl:param name="initial-font" rxml:type="font" select="
<xsl:param name="initial-font-size" rxml:type="int" sele
<xsl:param name="h2-fonts" rxml:type="string" select="'a

<xsl:import href="/templates/xhtmllayout.xsl" />
</xsl:stylesheet>
```

View template source

7.3 Customize Template

When focusing on XSL files, the button Customize template will be visible. This button will be available when XSL tags have certain parameters set in the template, making it possible to easily change the values of certain variables, e.g. color themes, in a simple wizard. How to write these parameters in the template is described in the Web Developer Manual.

8 Site Interaction

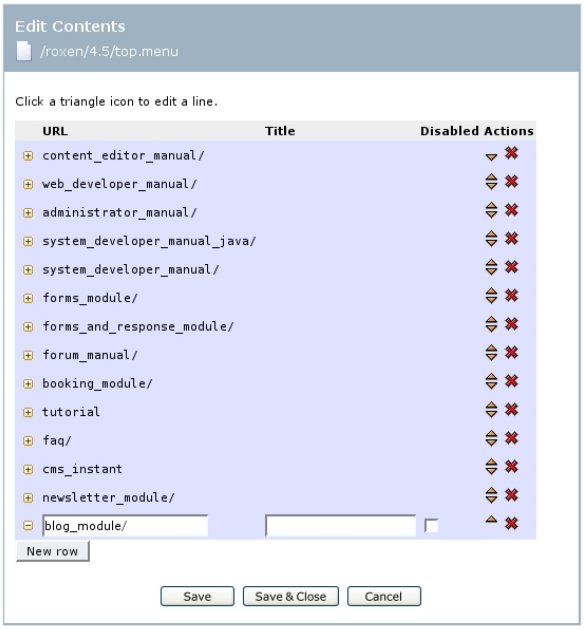
8.1 Menu Files

After the introduction of the new menu system in Roxen CMS 4.5 the older menu system described in this manual section remains for legacy reference purposes. The new menu system is described in the Insite Editor manual.

The legacy menu system can still be useful in certain circumstances in combination with emit#dir.

Menu files can be used to determine what will be shown in the navigation interface. A menu file has the file type Menu and might have the extension .menu, if extensions are used on the site.

A menu file is edited by focusing on the file and selecting the Edit content button in the Edit menu. A special web-based editor is used. A menu item consists of a page URL and a title that should be used in the navigation interface.



Edit Contents

/roxen/4.5/top.menu

Click a triangle icon to edit a line.

URL	Title	Disabled	Actions
⊞ content_editor_manual/			▽ ✖
⊞ web_developer_manual/			⇕ ✖
⊞ administrator_manual/			⇕ ✖
⊞ system_developer_manual_java/			⇕ ✖
⊞ system_developer_manual/			⇕ ✖
⊞ forms_module/			⇕ ✖
⊞ forms_and_response_module/			⇕ ✖
⊞ forum_manual/			⇕ ✖
⊞ booking_module/			⇕ ✖
⊞ tutorial			⇕ ✖
⊞ faq/			⇕ ✖
⊞ cms_instant			⇕ ✖
⊞ newsletter_module/			⇕ ✖
⊟ blog_module/		□	△ ✖

The menu editor

After entering the items it is possible to change the order they will be presented on the site by clicking on the arrows. Menu items may also be deleted by selecting the red-colored X to the right of each row. Save the changes of the menu file by clicking on the Ok button.

In case the URL is a relative URL to a file on the site it is not necessary to fill in the title field. The title from that file's metadata will be used instead. That way the title need only be changed once, in the metadata, and the change will take effect everywhere in the navigation interface.

Note!

> Access Control will be used to filter all menu items using relative URLs. That way the navigation interface will contain links to any pages the current user does not have access to, nor any broken links to pages that does not exist. If the menu file

is created before the files in the menu they will not show, until the files have been created.

Only relative URLs are filtered by access control. Absolute URLs will always be shown, regardless of permissions.

8.2 Directories and Files

The navigation menus can also be created by listing directories or files from a specified directory. When listing files it is possible to list only a certain type of files or files with a certain extension.

9 Version Control

The Version Control system is a key element for users of SiteBuilder, allowing for full control of all changes made to files in file structure. All versions of a file that have been committed to the repository are saved and can be viewed. It is also possible to revert to an earlier version of a file by simply clicking a button.

When files have been modified, one way or another, various symbols will be shown beside the file type symbol. The symbol in the first column, marked "You", represents the user's edit area, while the second, marked "Site", represents the file in the repository.

The significance of each of these can be interpreted somewhat differently depending on which column the symbol is shown in, and if it is shown in normal or undelete mode. See descriptions in the table below.

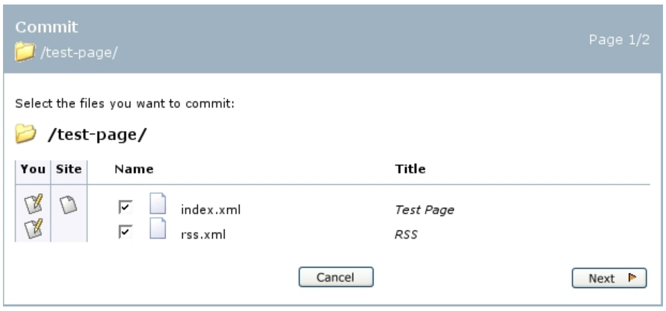
🗋		The file is created but has not been committed to the repository.
🗋	🗋	The file is copied to the user's edit area, but no changes have been saved.
🗋	📝	The file is copied to the user's edit area, but no changes have been saved, while another user has committed changes of the file to the repository.
🗋	☒	Current user has a copy of the file in the edit area. Another user has removed the file from the work area and has committed the action to the repository.
📝	🗋	The file has been modified and the changes have been saved in the user's edit area.
📝	📝	Another user has committed changes of the file to the repository. This indicates a possible conflict and an update will be necessary before the file can be committed.
📝	☒	Current user has saved changes to the file in the edit area. Another user has removed the file from the work area and has committed the action to the repository.
☒	🗋	The file has been removed but not committed. After being committed the file will only be seen in Undelete View.
☒	📝	The file has been removed but not committed. After being committed the file will only be seen in Undelete View.
	☒	A deleted file, as seen when in Undelete View.

9.1 Commit

When creating a new file or making changes to a file available to other SiteBuilder users, the file must be committed. Otherwise, the changes will only exist in the user's private edit area. When committed, the file will be saved in the repository where all SiteBuilder users with the right permissions can access it.

The repository is where a work area stores all files and all changes that have been committed. Even files that have been deleted are stored in the repository.

A file is committed by focusing on the file or the directory containing the file and clicking on the Commit button.



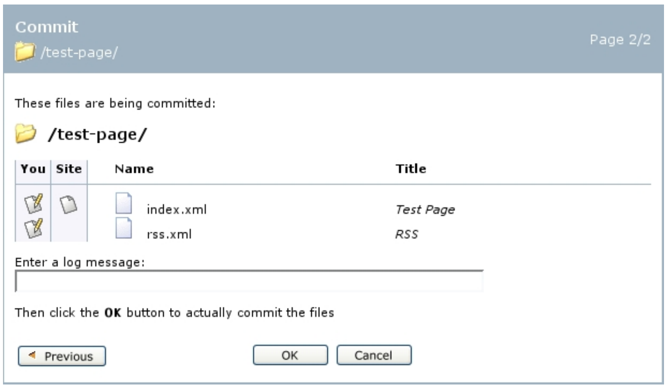
Select the files you want to commit:

/test-page/

You	Site		Name	Title
		☑	index.xml	*Test Page*
		☑	rss.xml	*RSS*

The commit wizard

Several files can be committed simultaneously by focusing on a directory. All changed files in the current directory will automatically be selected. To commit files in subdirectories they must be unfolded, which is done by clicking on the arrow icon in front of the directory name.



These files are being committed:

/test-page/

You	Site	Name	Title
		index.xml	*Test Page*
		rss.xml	*RSS*

Enter a log message:

Then click the **OK** button to actually commit the files

Commit wizard - log message

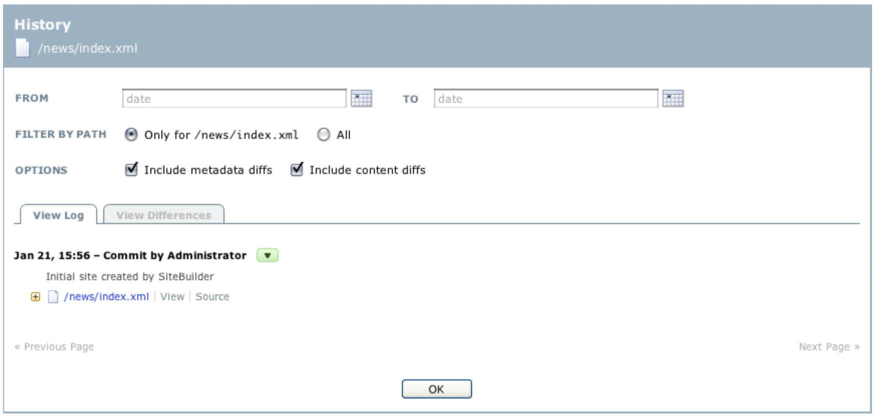
A log message must be provided to complete the commit wizard. If the messages are written carefully, they will be a valuable help when examining the logs.

If another user has committed changes of a file in the edit area, an update must be done to solve possible conflicts before the file can be committed.

9.2 History

The history wizards helps the users keep track of all committed versions of the files in SiteBuilder. Choosing History from the View menu when focusing on a file will create a wizard showing a list of all versions of the file and when and by whom they were committed together with the commit messages. When focusing on a directory the wizard will fetch data for all files in that directory recursively.

The wizard's functionality is extended when used together with the Labels feature. Below we will document its basic capabilities. See 9.3.3 for details about the label related functionality.



History
/news/index.xml

FROM date TO date

FILTER BY PATH ● Only for /news/index.xml ○ All

OPTIONS ☑ Include metadata diffs ☑ Include content diffs

Jan 21, 15:56 – Commit by Administrator

Initial site created by SiteBuilder

/news/index.xml | View | Source

The top part of the wizard contains filtering controls that are used to narrow down the history and help you find relevant information. The calendar icons give point-and-click access to entering dates into respective fields, though various date strings can be entered manually as well; if a date cannot be interpreted the entry field will signal an error with a red background color.

Any change to the filter will be reflected in a few seconds in the bottom half of the window. There are two tabs that give alternate views of the result:

View Log

Displays a reverse chronological history. All activities are grouped by commit actions and presented together with their commit message and author. Individual changes can be expanded to reveal metadata and content differences. Clicking the path itself will exit the wizard and navigate to the item’s location.

View Differences

Displays aggregated changes sorted by file name. Note that this tab is only enabled if the filter has been set to a date range.

In the Log view, each commit is accompanied by a popup menu, and each file entry has a View and/or Source links. The popup menu contains three actions:

Use as From

Sets the wizard’s From date filter to the date of the selected commit entry.

Use as To

Sets the wizard’s To date filter to the date of the selected commit entry.

Revert to Time...

Reverts a file or directory to the version in the given commit entry. Note that this is sensitive to the setting of the “Filter by path” control; if the scope is the full site the revert will apply to the site as a whole. You will be presented with a confirmation dialog before this action is executed.

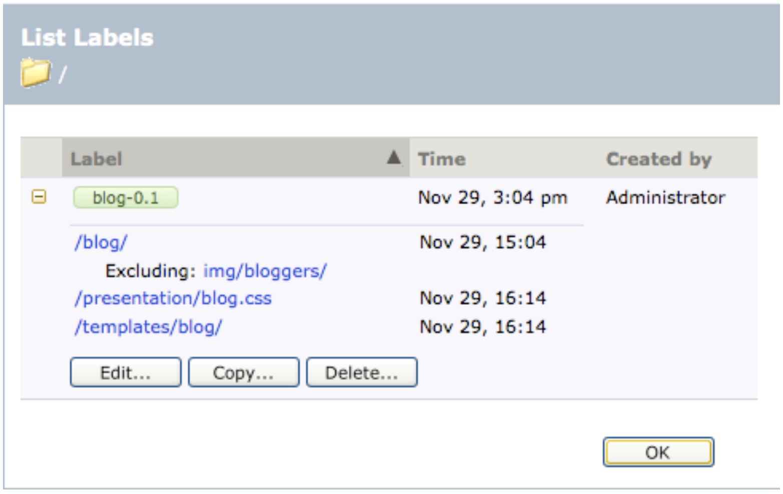
9.3 Labels

9.3.1 Introduction

The versioning system in Roxen CMS basically operates on the single file level: a file can be reverted to an earlier version; different versions of a file can be compared etc. By using labels, the possibilities of the versioning system are greatly enhanced. When labels are used, versioning can be applied not only to individual files but also to a directory, to a complete set of files and directories, and to the site as a whole. Please note: using the Label feature requires special licensing.

Labels are similar to tags in many other version control systems, i.e. a label records the state of the site at a specific point in time, to make it easier to keep track of events like releases, important edits, imports, etc.

One typical usage of labels is to group related files within a site, for instance because together they constitute a Blog function. This example Blog function may have some content in a directory called */blog/*, a few templates in */templates/blog/* and a separate stylesheet */presentation/blog.css*. When this set of files is labeled, versioning of the Blog functionality can be managed independently of other site parts.



List Labels

/

	Label ▲	Time	Created by
⊟	blog-0.1	Nov 29, 3:04 pm	Administrator
	/blog/	Nov 29, 15:04	
	Excluding: img/bloggers/		
	/presentation/blog.css	Nov 29, 16:14	
	/templates/blog/	Nov 29, 16:14	

The files and directories belonging to the label 'blog-0.1'.

A label can be viewed as a snapshot of a part of the site (or the whole site) at a specific point in time. When files and/or directories are labeled, the label is attached to their currently published state – the user local uncommitted edits are never part of a label.

The labels' timestamps are shown in minutes, but they are stored with complete accuracy, and they designate specific file versions even if there have been several commits within the same second.

All files and directories belonging to a label can be imported and exported as one package (see below for a detailed description). The label itself will also be transferred to the destination server. This is particularly useful when moving a new release of some function or site feature from a development/testing environment to production.

In order to create successive releases of a site or a part of it, a label can be copied. For instance, a label called *blog-0.1* may be attached to all the directories and files that constitute a Blog function, when someone first creates them. The same or other persons may then work on those files to prepare a first release of the Blog function for production. When the release is ready, the label can be copied to a new label called *blog-1.0*. This new label now contains the latest versions of all the files that constitute the Blog function, which is subsequently exported from the test environment to production. When newer versions of the Blog function are developed, they can be labeled in the same way, each time they are transferred to production. This makes it easy to upgrade and revert the entire function.

A label can be attached to a directory or to individual files. In the first case, all subdirectories and files within the labeled directory will inherit the label.

Sometimes one may want to exclude some part of a directory from the label. For example, the directory */blog/* may contain a directory *img/* which contains all images used in the layout of the blog pages. Let's say that this *img/* directory also contain a subdirectory *bloggers/* with portraits of all bloggers, and those pictures are maintained by the users in the production environment. We therefore want the *bloggers/* directory to be excluded from the label, so that blogger portraits will not be reset when a new release of the blog functionality is moved from development to production.

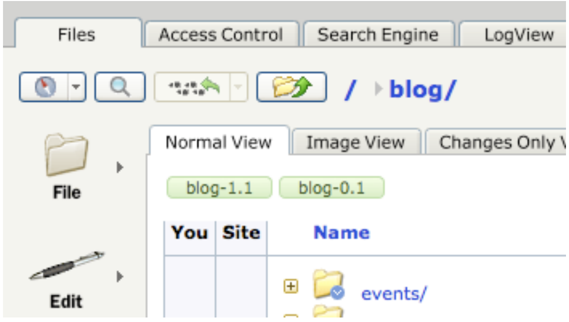
The use of labels for versioning purposes works best if the directory structure is well fitted to functional divisions of the site, so that different functionality is stored in separate files and directories.

Note!

> It is important to always have a clear view of which labels to use in the site and in what way. Labels can e.g. both be used to designate specific snapshots/releases, and collections of files (i.e. their timestamps are not significant).
>
> The labels will only show the state of affairs, but they cannot give any guidance as to what kind of release process is used. Therefore it is important to discuss and document the desired workflow and the corresponding use of labels among all those concerned.

9.3.2 File View

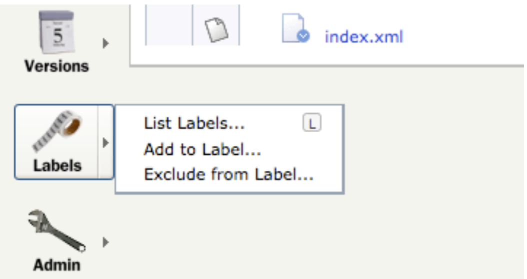
The main File view in the Content Editor shows a single line with the most recent labels that affect the current file or directory.



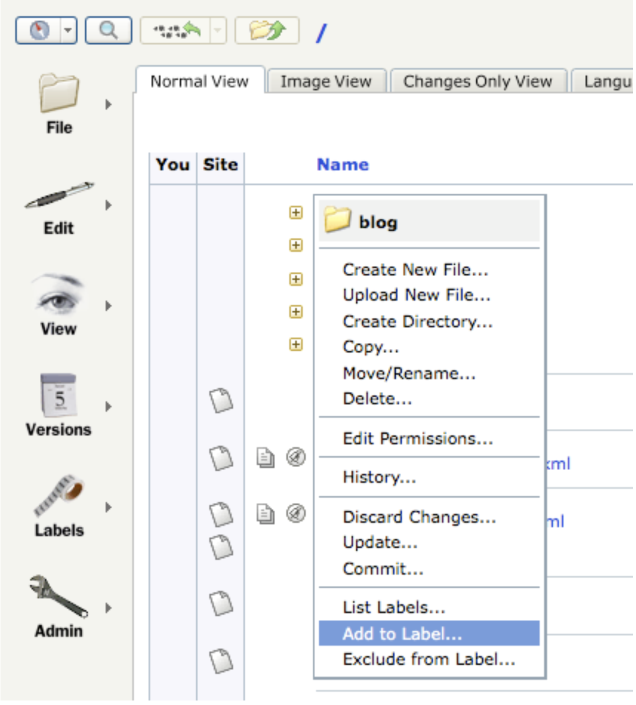
Labels affecting the current directory are shown in File view.

Menu

The Label feature is available from the left-hand menu and also from the context menu.



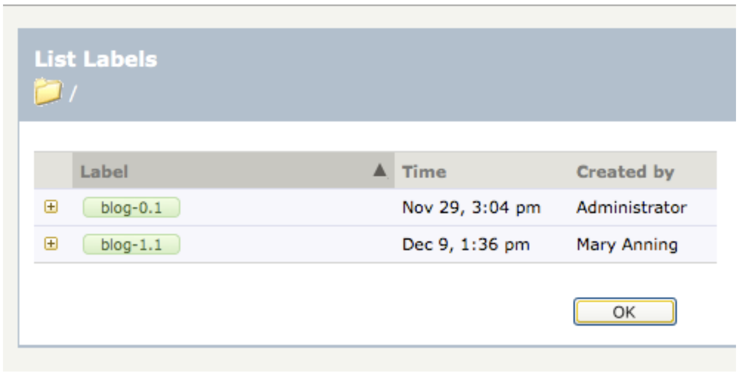
Left-hand menu.



Context menu.

List labels

A listing of labels with their associated files and directories can be obtained from any point in the site's file structure. Only the labels affecting the currently selected file or directory in the structure will be shown. To see all labels, list them from '/'.



List Labels

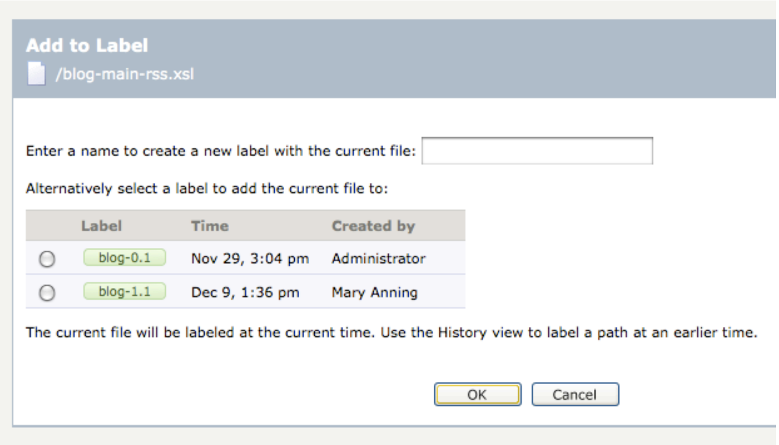
/

	Label	Time	Created by
⊞	blog-0.1	Nov 29, 3:04 pm	Administrator
⊞	blog-1.1	Dec 9, 1:36 pm	Mary Anning

List a label's files and directories.

Add to Label

Any file or directory can be added to a new or existing label. Only files that are committed can be added to a label.



Add to Label

/blog-main-rss.xsl

Enter a name to create a new label with the current file:

Alternatively select a label to add the current file to:

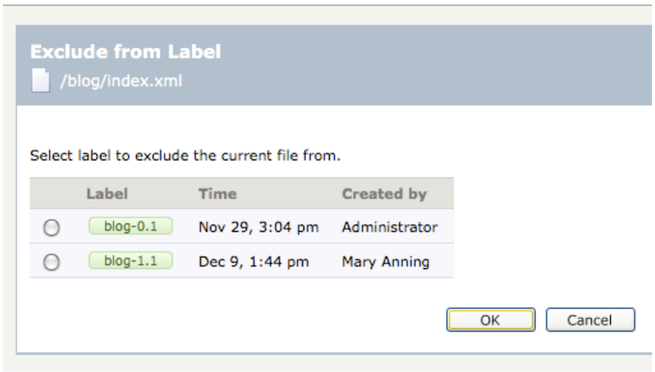
	Label	Time	Created by
○	blog-0.1	Nov 29, 3:04 pm	Administrator
○	blog-1.1	Dec 9, 1:36 pm	Mary Anning

The current file will be labeled at the current time. Use the History view to label a path at an earlier time.

Add a file to a label.

Exclude from Label

Any file or directory in a label can be excluded from that label. Excluding objects is also available from the Edit label functionality (see below).



Exclude from Label

/blog/index.xml

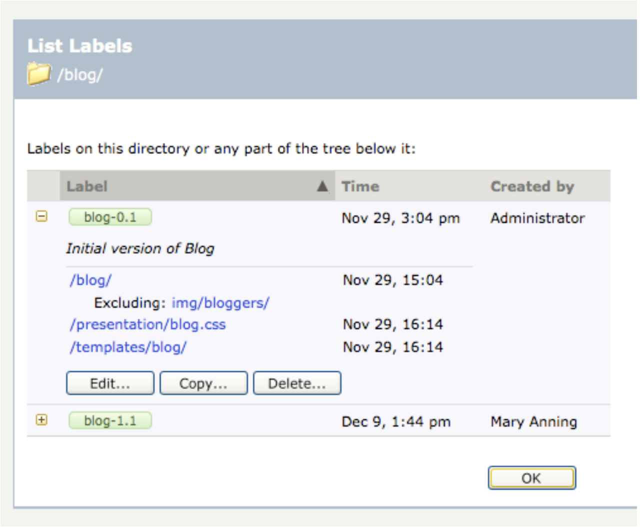
Select label to exclude the current file from.

	Label	Time	Created by
○	blog-0.1	Nov 29, 3:04 pm	Administrator
○	blog-1.1	Dec 9, 1:44 pm	Mary Anning

Exclude a file from a label.

Edit, Copy, Delete

A label can be edited, copied and deleted. The functions Edit, Copy and Delete are available from the "List label" feature and also as a dropdown from the label's name in the main File view.



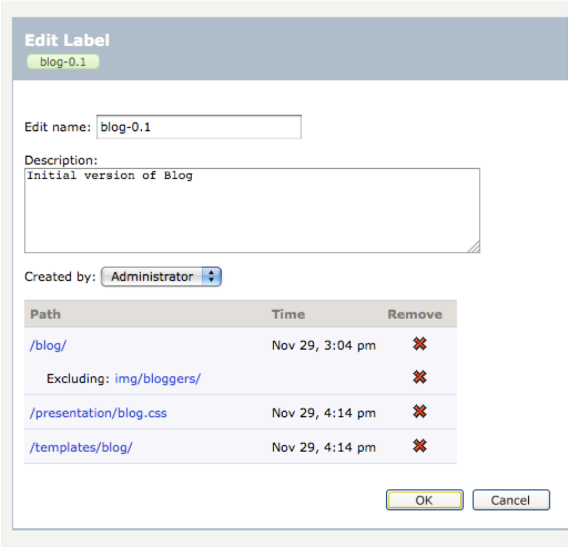
List Labels

/blog/

Labels on this directory or any part of the tree below it:

Label	Time	Created by
⊟ blog-0.1	Nov 29, 3:04 pm	Administrator
Initial version of Blog		
/blog/	Nov 29, 15:04	
Excluding: img/bloggers/		
/presentation/blog.css	Nov 29, 16:14	
/templates/blog/	Nov 29, 16:14	
Edit... Copy... Delete...		
⊞ blog-1.1	Dec 9, 1:44 pm	Mary Anning

The functions Edit, Copy and Delete in the "List label" feature.



Edit Label

blog-0.1

Edit name: blog-0.1

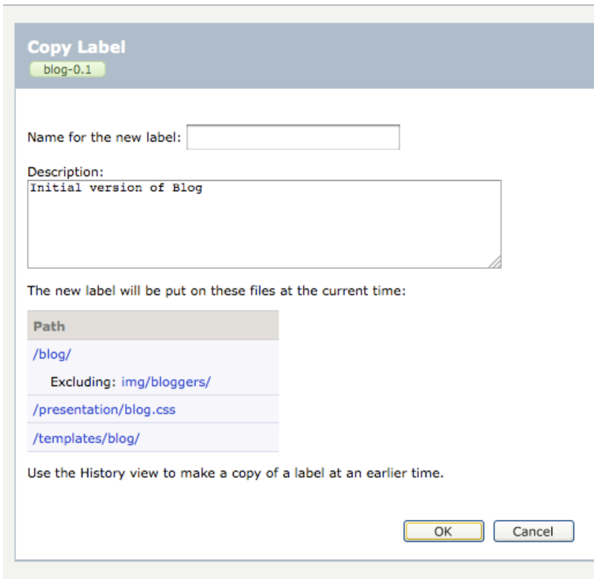
Description:

```
Initial version of Blog
```

Created by: Administrator

Path	Time	Remove
/blog/	Nov 29, 3:04 pm	✖
Excluding: img/bloggers/		✖
/presentation/blog.css	Nov 29, 4:14 pm	✖
/templates/blog/	Nov 29, 4:14 pm	✖

Edit the name, description, and creator of a label; remove objects from the label.



Copy Label

blog-0.1

Name for the new label:

Description:

```
Initial version of Blog
```

The new label will be put on these files at the current time:

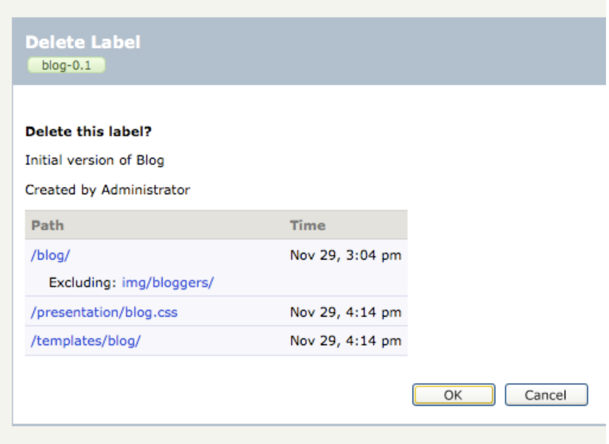
Path
/blog/ Excluding: img/bloggers/
/presentation/blog.css
/templates/blog/

Use the History view to make a copy of a label at an earlier time.

Copy an existing label.

When a label is copied, the new label will contain the same files and directories as the original label, but with their latest versions.

An entire label can be deleted. When a label is to be deleted, the Content Editor will ask confirmation first. The files and directories belonging to the label will not be deleted.



Delete Label

blog-0.1

Delete this label?

Initial version of Blog

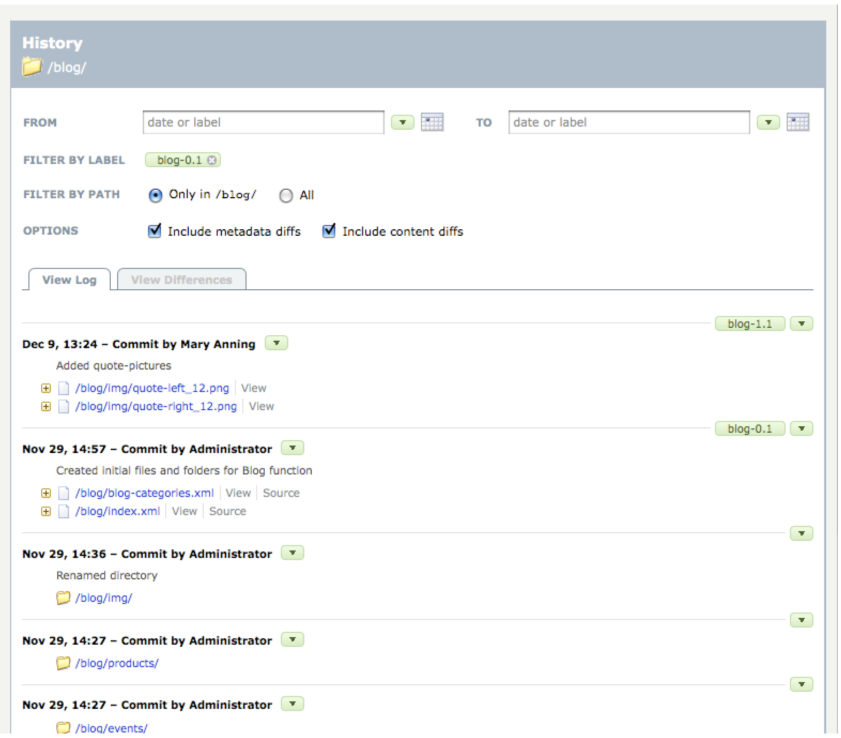
Created by Administrator

Path	Time
/blog/ Excluding: img/bloggers/	Nov 29, 3:04 pm
/presentation/blog.css	Nov 29, 4:14 pm
/templates/blog/	Nov 29, 4:14 pm

Delete a label.

9.3.3 History View

The History view shows a listing of all versions of a selected file, or of files and subdirectories within a selected directory. It also shows which versions of files belong to which label.



History
/blog/

FROM date or label TO date or label

FILTER BY LABEL blog-0.1

FILTER BY PATH ● Only in /blog/ ○ All

OPTIONS ☑ Include metadata diffs ☑ Include content diffs

blog-1.1

Dec 9, 13:24 – Commit by Mary Anning

Added quote-pictures

/blog/img/quote-left_12.png | View
/blog/img/quote-right_12.png | View

blog-0.1

Nov 29, 14:57 – Commit by Administrator

Created initial files and folders for Blog function

/blog/blog-categories.xml | View | Source
/blog/index.xml | View | Source

Nov 29, 14:36 – Commit by Administrator

Renamed directory

/blog/img/

Nov 29, 14:27 – Commit by Administrator

/blog/products/

Nov 29, 14:27 – Commit by Administrator

/blog/events/

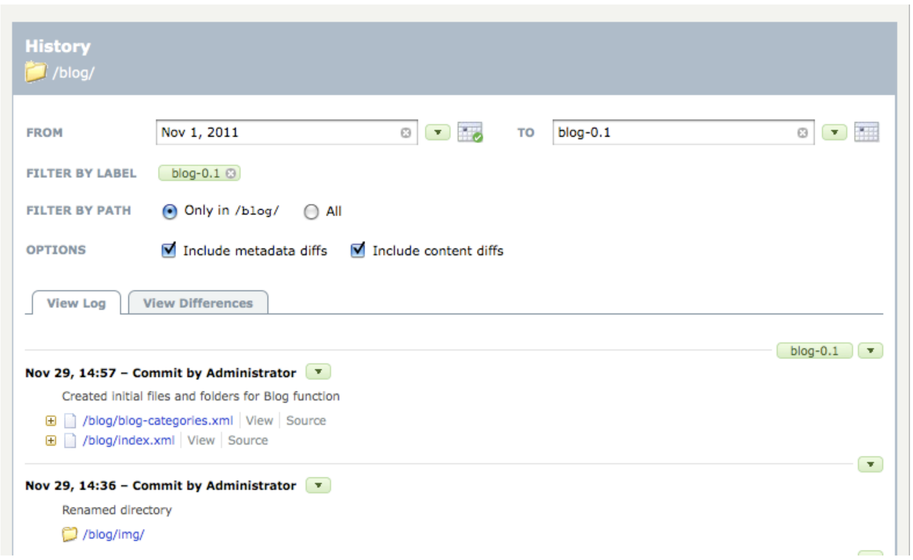
Labels are shown in the History view.

Time range

In the History view, a time range can be specified (*From – To*), using either a date or a label. If a label is used, its timestamp will be used as the *From* or *To* date. This is particularly useful to see the changes from one release to another, or the combined differences between them under the View Differences tab. (Please note that View Differences is only available when a time range is specified.)

Choosing a *From* label means that only commits *after* this label will be shown in the History view. Therefore, if you choose for instance label *blog-0.1* as *From* you will not see the labelled commits themselves, but only those after the time of labeling. This is useful to see what happened after a label, e.g. up to the next labelled version if you choose that label as *To*.

Note that if a date or time is chosen in the *From* field instead of a label, the results are different. In that case you will see all commits from and including that date or time. A date or time in the *To* field is inclusive as well, so if you e.g. choose 2010-10-10 in both *From* and *To*, you will see all commits during that day.



History
/blog/

FROM: Nov 1, 2011
TO: blog-0.1
FILTER BY LABEL: blog-0.1
FILTER BY PATH: ● Only in `/blog/` ○ All
OPTIONS: ☑ Include metadata diffs ☑ Include content diffs

Nov 29, 14:57 – Commit by Administrator
Created initial files and folders for Blog function
/blog/blog-categories.xml | View | Source
/blog/index.xml | View | Source

Nov 29, 14:36 – Commit by Administrator
Renamed directory
/blog/img/

A time range for the History view is specified using a date for From *and a label for* To*.*

Filter by Label

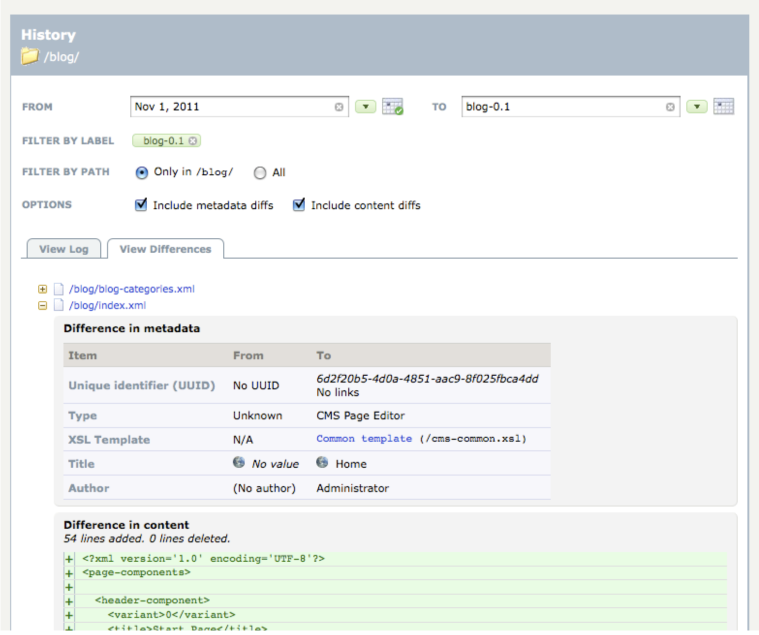
If a label is used in the *From* or *To* fields in the History view, it is automatically used as a file selection filter as well. The name of the label shows up as *Filter by Label*. Now, only the files that belong to the label are listed in the History view. The label selection filter can be removed to reveal all files that happen to fall into the time range selected by the labels' timestamps.

When filtering by two labels at once, all files covered by either of those labels are shown.

View Differences

The View Differences tab shows the combined changes between the two times in the *From* and *To* fields. If you choose two labels as *From* and *To*, they work together with *Filter by Label* so that you see the combined changes between those labels, and only for the files covered by them. This is particularly useful when the labels are for different versions of the same file selection.

So given the example with Blog function described earlier, to see all changes in the Blog related files between version 0.1 and 1.0, you would specify the label *blog-0.1* as *From* and *blog-1.0* as *To*. Due to the *Filter by Label* function, the view is not cluttered with changes in other parts of the site that were made during the same period.



History
/blog/

FROM: Nov 1, 2011 TO: blog-0.1

FILTER BY LABEL: blog-0.1

FILTER BY PATH: ● Only in /blog/ ○ All

OPTIONS: ☑ Include metadata diffs ☑ Include content diffs

/blog/blog-categories.xml
/blog/index.xml

Difference in metadata

Item	From	To
Unique identifier (UUID)	No UUID	*6d2f20b5-4d0a-4851-aac9-8f025fbca4dd* No links
Type	Unknown	CMS Page Editor
XSL Template	N/A	Common template (/cms-common.xsl)
Title	*No value*	Home
Author	(No author)	Administrator

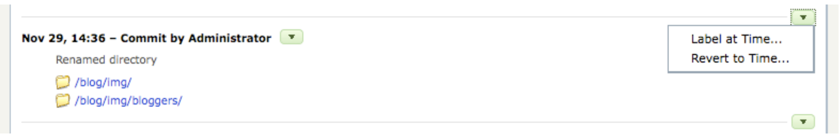
Difference in content
54 lines added. 0 lines deleted.

```
+ <?xml version='1.0' encoding='UTF-8'?>
+ <page-components>
+
+   <header-component>
+     <variant>0</variant>
+     <title>Start Page</title>
```

View differences.

Revert

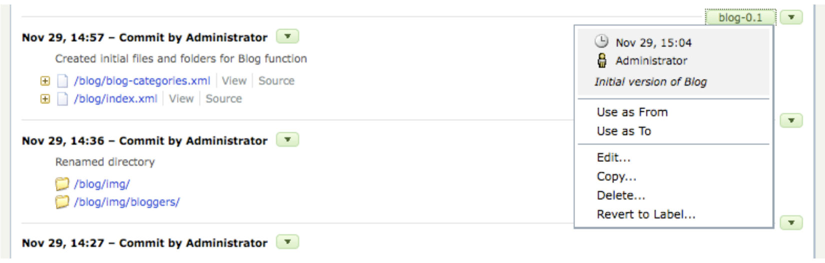
The History View provides several different ways to revert to previous versions. Revert to Time reverts all files currently included in the History View to the time between two commits. I.e. it only acts on the directory or file the History View was opened from, and if there are one or more labels in the *Filter by Label* filter, only the files below the directory which are included in either of those labels are subject to the revert action.



Nov 29, 14:36 – Commit by Administrator
Renamed directory
/blog/img/
/blog/img/bloggers/

Revert to Time.

Revert to Label reverts all files in a label to the “snapshot” state they have in that label. Note that this action is *not* limited to the directory the History View was opened from.



Nov 29, 14:57 – Commit by Administrator
Created initial files and folders for Blog function
/blog/blog-categories.xml | View | Source
/blog/index.xml | View | Source

Nov 29, 14:36 – Commit by Administrator
Renamed directory
/blog/img/
/blog/img/bloggers/

Nov 29, 14:27 – Commit by Administrator

blog-0.1
Nov 29, 15:04
Administrator
Initial version of Blog

Revert to Label.

Note!

> If Revert to Time is done in the History View while '/' is selected, the whole site is reverted to the specified time.

Creating backdated labels

The History view can be used to create backdated labels. In other words, to add arbitrarily old versions of files and directories to a new or existing label. Just like the revert actions described above, it has two variants:

Label at Time creates a label at the time between two commits, and the new label covers all files and directories included in the file selection in the History View. I.e. the label is put on the file or directory where the History View was invoked, and if *Filter by Label* is in use, it is further restricted to the files and directories included in any of those labels.

Note!

> There is no way to specify the exact time between two commits where the label is created. That is because it doesn't matter – a label created at any time in that time interval would be identical, since if there were some other commit that would affect it, that commit would be listed between the two other ones.

Label Commit from the context menu for a specific commit creates a label from the file (or files) included in that commit, at the committed versions.

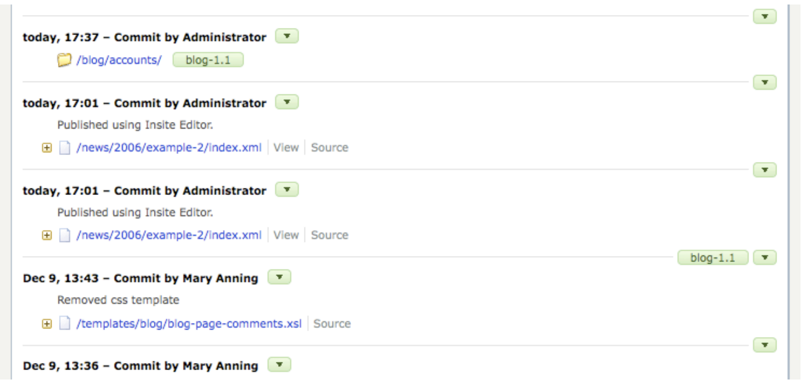
Note!

> Even if a commit committed all changes in a whole directory tree, the Label Commit action still labels all the committed files (and any created or deleted directories) individually. This means that the label will contain a potentially long list of files instead of the directory and the commit time. That makes the label more cluttered, and it also affects how Export and Import of the label works (see the note in 9.3.4). It is therefore normally better to use Label at Time.

Labels with dispersed timestamps

Once a label is in place, a newer file or directory can also be added to that label by using the Label Commit action. This means that the file version in the selected commit is made part of the label. That version then takes precedence over the timestamp the file already may have in the label. If there are several files in the same commit then all of them are added to the label in the same way.

When a later commit has been added to a label, the label shows up right next to each file in that commit. The label is also still shown on its original time line to the right of the History View page (which is called the *baseline time* of the label).



today, 17:37 – Commit by Administrator ▼

/blog/accounts/ blog-1.1

today, 17:01 – Commit by Administrator ▼

Published using Insite Editor.

⊞ /news/2006/example-2/index.xml | View | Source

today, 17:01 – Commit by Administrator ▼

Published using Insite Editor.

⊞ /news/2006/example-2/index.xml | View | Source

blog-1.1 ▼

Dec 9, 13:43 – Commit by Mary Anning ▼

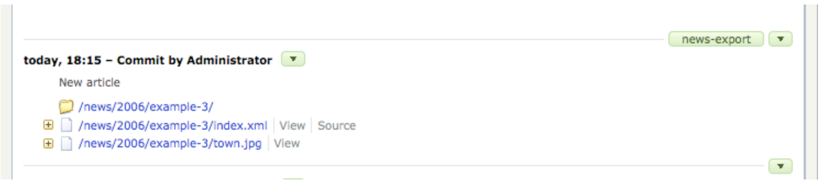
Removed css template

⊞ /templates/blog/blog-page-comments.xsl | Source

Dec 9, 13:36 – Commit by Mary Anning ▼

History View showing that the commit of /blog/accounts/ *was added today to the existing label* blog-1.1*. The label also shows up on its original timeline (to the right of the page).*

It is also possible to add an *earlier* commit to a label. The History View then takes the time of the old commit as the baseline time of the label, so the label is shown at that point in time to the right of the page. The newer versions that were already present in the label are now labeled individually, just as if they were added afterwards.



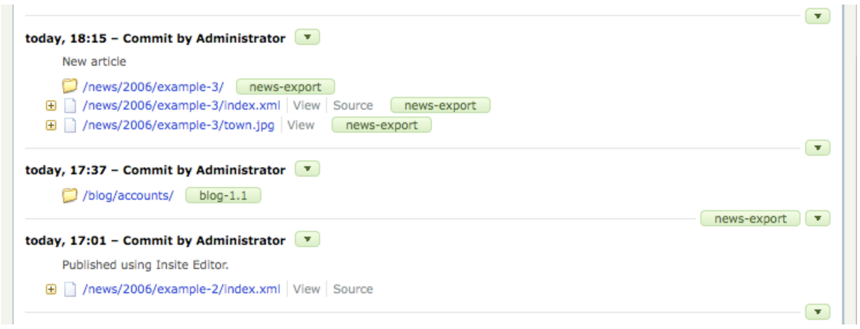
news-export

today, 18:15 – Commit by Administrator

New article

/news/2006/example-3/
/news/2006/example-3/index.xml | View | Source
/news/2006/example-3/town.jpg | View

A label news-export *has been added to the directory* /news/2006/example-3/ *today at 18:15.*



today, 18:15 – Commit by Administrator

New article

/news/2006/example-3/ news-export
/news/2006/example-3/index.xml | View | Source news-export
/news/2006/example-3/town.jpg | View news-export

today, 17:37 – Commit by Administrator

/blog/accounts/ blog-1.1

news-export

today, 17:01 – Commit by Administrator

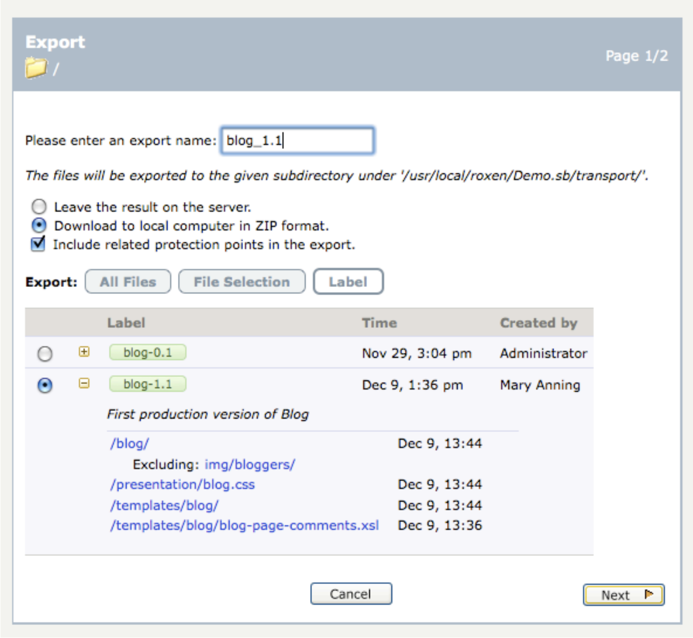
Published using Insite Editor.

/news/2006/example-2/index.xml | View | Source

An earlier commit (today 17:01) has been added to the news-export *label. The earlier commit is now the baseline time for the label, and the versions that were already present in the label are now labeled individually.*

9.3.4 Export and Import

The Export and Import actions have been extended to handle labels as well. A label can be chosen in the Export wizard. Then only the files included in that label are exported, and furthermore the versions of those files that are specified by the label.



Please enter an export name: blog_1.1

The files will be exported to the given subdirectory under '/usr/local/roxen/Demo.sb/transport/'.

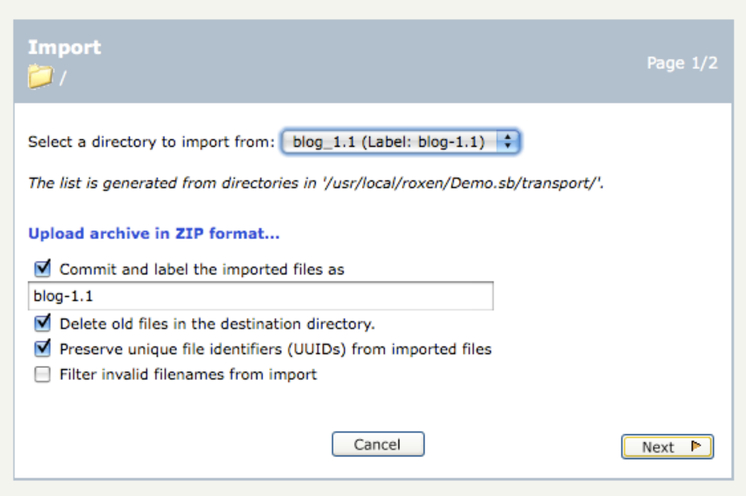
○ Leave the result on the server.
● Download to local computer in ZIP format.
☑ Include related protection points in the export.

Export: All Files | File Selection | Label

		Label	Time	Created by
○	⊞	blog-0.1	Nov 29, 3:04 pm	Administrator
●	⊟	blog-1.1	Dec 9, 1:36 pm	Mary Anning
		First production version of Blog		
		/blog/	Dec 9, 13:44	
		Excluding: img/bloggers/		
		/presentation/blog.css	Dec 9, 13:44	
		/templates/blog/	Dec 9, 13:44	
		/templates/blog/blog-page-comments.xsl	Dec 9, 13:36	

Export Label.

The label itself is included in the export archive, so when it is chosen in the Import wizard, there is an option to recreate the label with the same name and description.



Select a directory to import from: blog_1.1 (Label: blog-1.1)

The list is generated from directories in '/usr/local/roxen/Demo.sb/transport/'.

Upload archive in ZIP format...

☑ Commit and label the imported files as
blog-1.1
☑ Delete old files in the destination directory.
☑ Preserve unique file identifiers (UUIDs) from imported files
□ Filter invalid filenames from import

Import Label.

In this case the imported files and directories are immediately committed and added to the imported label, thus making sure it covers identical content as on the export server. This is useful to create file packages and transfer them between servers, and ensure that the labels remain consistent between all servers.

Note!

> With export/import there is a very important distinction between a labeled directory and labeled files within a directory. If the directory itself is labeled, an exact copy will be transferred. This means that files which have been deleted on the export server will also be deleted on the import server. In the same way, files that have never existed on the export server, but were added to the labeled folder

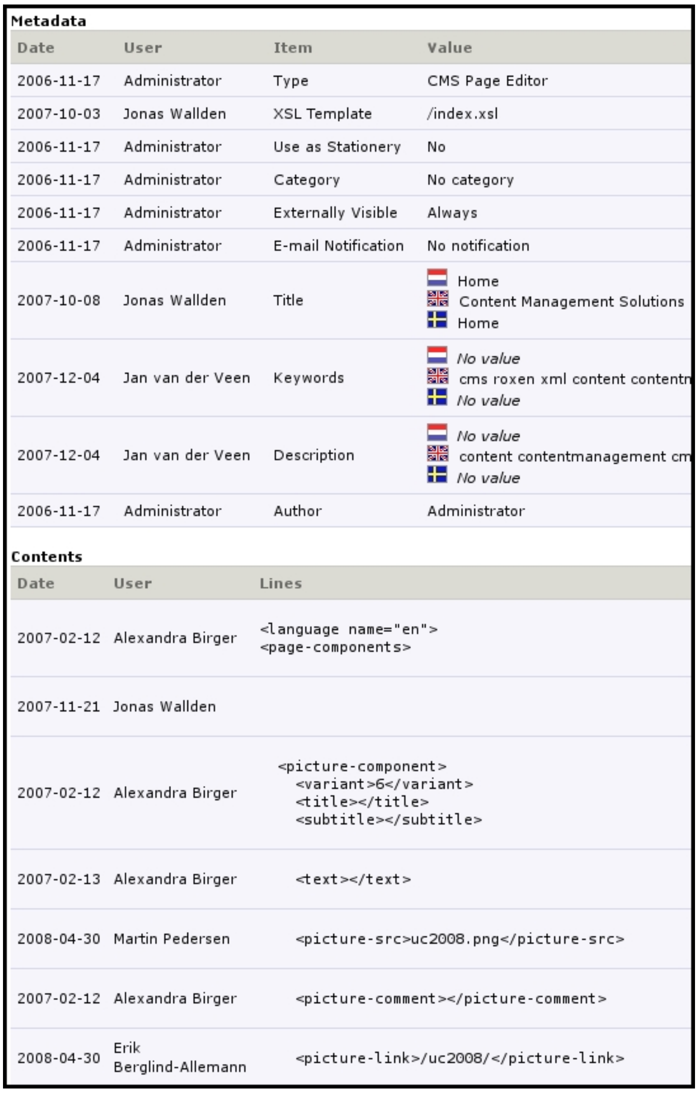
on the destination server, will also be deleted. This ensures that the same label will have exactly the same content on all servers.

If, on the other hand, individual files within a directory are labeled, newly deleted or absent files on the export server will *not* be deleted on the import server. Similarly, within a labeled directory, files or directories which have been excluded from that label, will not be part of the export and will not be overwritten nor removed from the destination server.

In general, it is more efficient to label directories instead of individual files, and to keep the exclusion of objects from a label to a minimum. When devising a site (or site feature), some thought should go into the best directory structure that allows for an optimal use of labeling.

9.4 Contributors

When focusing a file select Contributors from the View menu to produce a wizard showing all changes made to the metadata and contents of a file and when and by whom they were committed.



Metadata

Date	User	Item	Value
2006-11-17	Administrator	Type	CMS Page Editor
2007-10-03	Jonas Wallden	XSL Template	/index.xsl
2006-11-17	Administrator	Use as Stationery	No
2006-11-17	Administrator	Category	No category
2006-11-17	Administrator	Externally Visible	Always
2006-11-17	Administrator	E-mail Notification	No notification
2007-10-08	Jonas Wallden	Title	🇳🇱 Home 🇬🇧 Content Management Solutions 🇸🇪 Home
2007-12-04	Jan van der Veen	Keywords	🇳🇱 *No value* 🇬🇧 cms roxen xml content contentn 🇸🇪 *No value*
2007-12-04	Jan van der Veen	Description	🇳🇱 *No value* 🇬🇧 content contentmanagement cm 🇸🇪 *No value*
2006-11-17	Administrator	Author	Administrator

Contents

Date	User	Lines

2007-02-12	Alexandra Birger	`<language name="en">`
		`<page-components>`

2007-11-21	Jonas Wallden	

2007-02-12	Alexandra Birger	`<picture-component>`
		`  <variant>6</variant>`
		`  <title></title>`
		`  <subtitle></subtitle>`

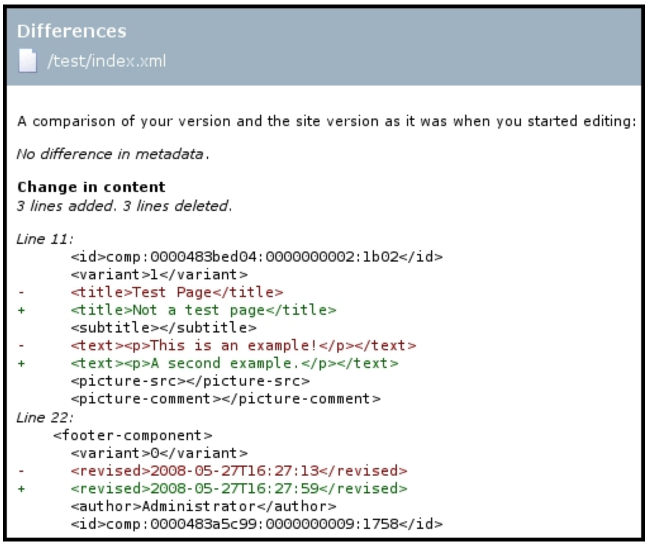
2007-02-13	Alexandra Birger	`<text></text>`
2008-04-30	Martin Pedersen	`<picture-src>uc2008.png</picture-src>`
2007-02-12	Alexandra Birger	`<picture-comment></picture-comment>`
2008-04-30	Erik Berglind-Allemann	`<picture-link>/uc2008/</picture-link>`

The contributors pane

9.5 Diff

When having changed a file the user sometimes want to see the differences from the version in the repository. This is especially useful when you have to update a file when another user has been committing changes while you are editing the file.

While focusing on the file select the Diff button in the View menu. The parts of the file that are the same in both the repository and your edit area are shown in black. Lines that are added by the user have a "+" in front and are shown in green. Lines that are added by another user or deleted by you have a "-" in front and are shown in red.



Differences

/test/index.xml

A comparison of your version and the site version as it was when you started editing:

No difference in metadata.

Change in content
3 lines added. 3 lines deleted.

```
Line 11:
        <id>comp:0000483bed04:0000000002:1b02</id>
        <variant>1</variant>
-       <title>Test Page</title>
+       <title>Not a test page</title>
        <subtitle></subtitle>
-       <text><p>This is an example!</p></text>
+       <text><p>A second example.</p></text>
        <picture-src></picture-src>
        <picture-comment></picture-comment>
Line 22:
      <footer-component>
        <variant>0</variant>
-       <revised>2008-05-27T16:27:13</revised>
+       <revised>2008-05-27T16:27:59</revised>
        <author>Administrator</author>
        <id>comp:0000483a5c99:0000000009:1758</id>
```

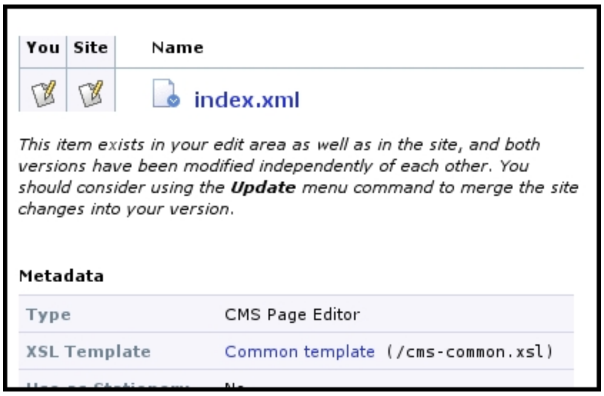
The diff page

9.6 Editing Status

The editing status wizard shows the status all copies of the focused file existing in the edit areas of different users.

9.7 Update

When a user is editing a file and another user commits changes of that file to the repository, the status icons file will indicate a conflict as shown in the figure below.



You	Site	Name
		index.xml

This item exists in your edit area as well as in the site, and both versions have been modified independently of each other. You should consider using the ***Update*** *menu command to merge the site changes into your version.*

Metadata

Type	CMS Page Editor
XSL Template	Common template (/cms-common.xsl)

File indicating conflict

Before the file can be committed it must be updated with the latest version from the repository. While updating a file, a number of conflicts and differences may arise, the different types are described below the headings Conflicts and Differences below.

9.7.1 Solving Conflicts

When focusing on a file it can be updated by clicking on Update. When clicking on Next the wizard searches for conflicts between the copy in the edit area and the one in the repository.

When focusing a directory, the Update wizard will produce a file listing. After selecting files for update, clicking on Next will continue to the File status summary which gives the present status of each file selected.

If the wizard finds several files with conflicts, each file has to be resolved separately before they can be updated. To resolve a conflict, one of the files must be chosen from the list. Files which only contain differences and no conflicts do not need resolving but can be updated directly.

9.7.2 Conflicts

Different content types

The content types in the files are different. This conflict must be solved before any of the other conflicts can be handled, because the methods used for solving conflicts depend on the content type of the file. When a content type conflict is resolved, another conflict might arise that is due to something else.

Unresolved conflicts

The same part is changed in both files.

9.7.3 Differences

Conflict-free differences

For instance:

- New data is added.
- Old data is removed.
- Old data is replaced by new data.

9.7.4 Status

No differences

Both files are identical.

Resolved conflicts

All conflicts are solved.

Removed from site

File or directory is removed from the site.

9.7.5 Actions

Add your file

Replace the present version in the repository. The replaced version will still be available in the log due to the version control system.

Discard your file

Remove the file from the edit area. All local changes will be lost permanently.

9.7.6 Navigation Help

To help the user navigate through the wizard, words and different colors have been used to represent different aspects:

Red

Represents a conflict and is found in the text to highlight a conflict and to the left in the wizard.

Green

Represents the edit area. Text on green background does only exist in the edit area.

Blue

Represents the site repository. Text on blue background has been added by another user and does not exist in the edit area.

Gray

Represents old value or content. When defined as old, it is not considered to be a conflict, only a difference between the files.

Old

The word Old is set beside an unselected radio button. It is always represented by gray color.

Del

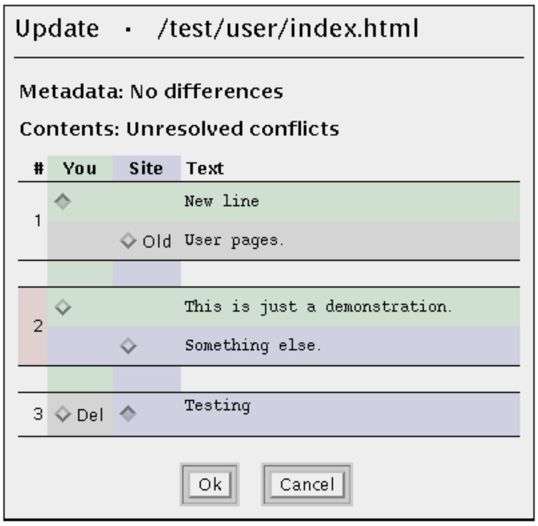
The word Del is set beside an unselected radio button. Del implies that the data only exists in one file.

Radio buttons

A preselected radio button implies that this is the probable choice the user will make. The update wizard always assumes the latest changes should be kept, so these radio buttons are preselected.

The radio button should be selected in front of data that is to be kept and the Del button in front of data that is to be removed. Data with only an unselected radio button and no Del button in front will be deleted.

By selecting a file, the wizard shows the conflict resolving mode. Here, the changes in a file necessary to make an update are made by choosing what data should be kept and what should be removed.



Update · /test/user/index.html

Metadata: No differences

Contents: Unresolved conflicts

#	You	Site	Text
1	◇		New line
		◇ Old	User pages.
2	◇		This is just a demonstration.
		◇	Something else.
3	◇ Del	◇	Testing

Update example showing conflict

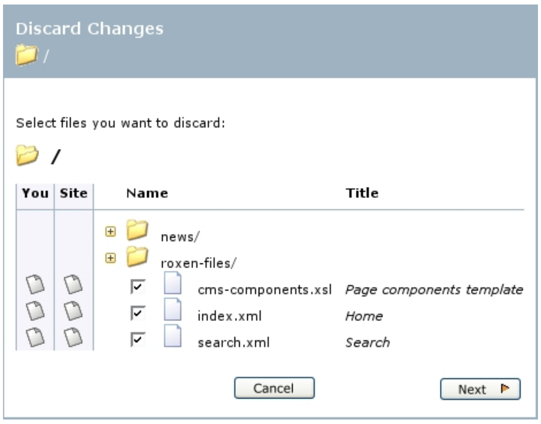
This example shows a conflict where two users have changed the same part of the content. The radio button in front of the text should be selected for the text to be kept. When the proper radio buttons are selected, choose Ok to update the file.

9.7.7 After Solving

If several files need to be updated and not all files are conflict free, an update is still possible since the conflict free files will be updated. When all updates are finished, the files can be committed.

9.8 Discard Changes

Changes made to a file since the file was last committed can be canceled by choosing Discard your changes from the menu. Just remember that the changes will be lost permanently.



Discard Changes

/

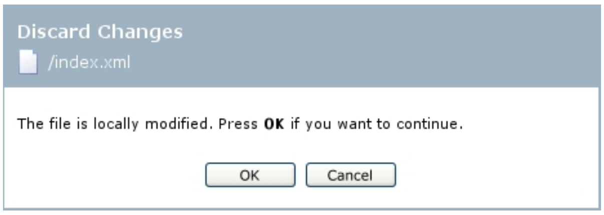
Select files you want to discard:

/

You	Site		Name	Title
		⊞	news/	
		⊞	roxen-files/	
		☑	cms-components.xsl	*Page components template*
		☑	index.xml	*Home*
		☑	search.xml	*Search*

The discard wizard

If a file is changed or if you discard a file not checked into the repository, the wizard will ask for confirmation before removing the copy from the edit area.



Discard Changes

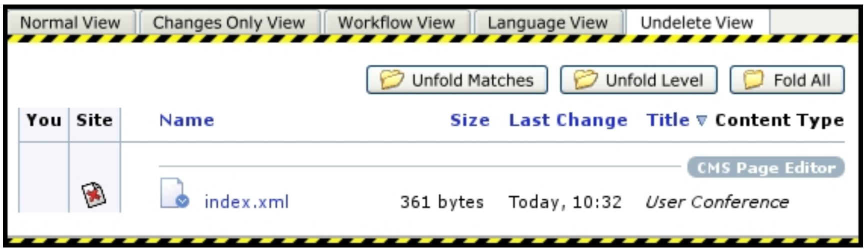
/index.xml

The file is locally modified. Press **OK** if you want to continue.

Confirming the discard action

9.9 Undelete

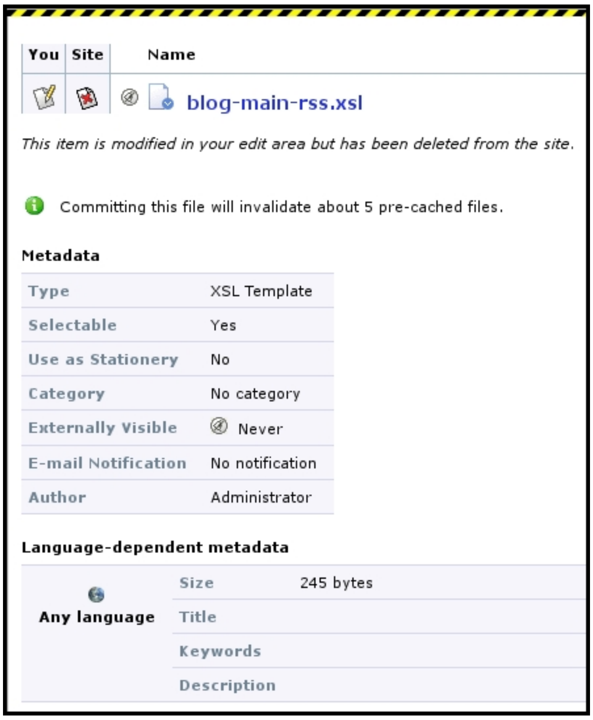
A file that has been committed to the repository will be stored forever and if deleted it can be restored, or undeleted. Deleted files can be seen in the file system after clicking on the Undelete View tab.



You	Site	Name	Size	Last Change	Title ▿	Content Type
		index.xml	361 bytes	Today, 10:32	*User Conference*	

Undelete View

When focusing on a deleted file the View options and the Undelete button will be available. All versions of the file can be viewed and restored. Clicking on the Undelete button will restore the focused file.



You	Site	Name
		blog-main-rss.xsl

This item is modified in your edit area but has been deleted from the site.

Committing this file will invalidate about 5 pre-cached files.

Metadata

Type	XSL Template
Selectable	Yes
Use as Stationery	No
Category	No category
Externally Visible	Never
E-mail Notification	No notification
Author	Administrator

Language-dependent metadata

Any language	Size	245 bytes
	Title	
	Keywords	
	Description	

Undeleted file

The undeleted file must then be committed, which is only possible after leaving Undelete View, before other users will be able to see it.

9.9.1 Unrestorable Files

To be able to restore a deleted file it must be stored in the repository of the Version Control system. A file that was deleted before being committed or deleted with Purge consequently will not be found when entering Undelete View.

10 Work Areas

This chapter focuses on how to handle work areas. The concept of using multiple work areas comes from needing several copies of a live site functioning as advanced test versions of the live site. This makes it possible for the user and designer of the site to enter and evaluate content and layout before publishing it on the live site, without being afraid of causing damage.

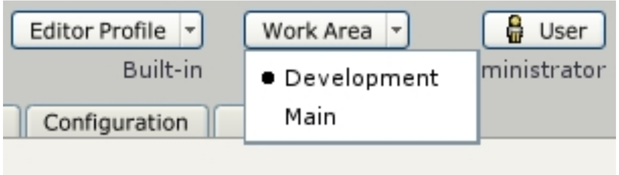
Only a part of the work area functionality is explained here because the functionality is divided between the administrator and the user. The configuration of a work area is left to the administrator. The part that considers the user is the Work Area select box in the header and Join under the Files tab.

For the join function to work properly, two work areas must never differ too much. A second work area should only be used as a temporary version of the live site, used for testing or evaluating changes. Use the join function to merge the different work areas when the testing is done.

10.1 Changing Work Area

The current work area is shown in the upper right corner in the SiteBuilder Editor, below the user name.

Sometimes two work areas are used, one containing the live site (by default this work area is called Main) and the other containing a test version. This is quite useful when several users are working together on advanced changes of the site.



Work area select widget

To change work area click on the Work Area button to go to the work area select wizard. Choose the proper work area from the select box and click on Ok to confirm your choice.

10.2 Join

Using two work areas is useful when you want to work completely separate from the live site and have even better control over all changes on the site before launching it onto the world wide web. When the site is ready to be launched all the data must be transferred to the live site. This is done by using the Join work areas button (Versions menu) under the Files tab.

By joining two work areas you merge the two versions of the same file or directory from each work area with each other. You cannot join files that have copies in your edit area, they must be committed first.

10.2.1 Solving Problems

The wizard begins by asking you to choose which work area to join your focused file or directory with. By clicking on Next the wizard searches for conflicts in those files. The different types of conflicts and differences you may encounter are described below:

10.2.2 Conflicts

Different content types

The content types in the files are different. This conflict must be solved before any of the other conflicts can be handled, because the methods used for solving conflicts depend on the content type of the file. When a content type conflict is resolved a new conflict might arise that is due to something else.

Unresolved conflicts

The same part is changed in both files.

10.2.3 Differences

Conflict-free differences

For instance:

- New data is added.
- Old data is removed.
- Old data is replaced by new data.

Doesn't exist in work area

File exists in one work area but not in the other.

10.2.4 Status

No differences

Both files are identical.

Resolved conflicts

All conflicts are solved.

10.2.5 Actions

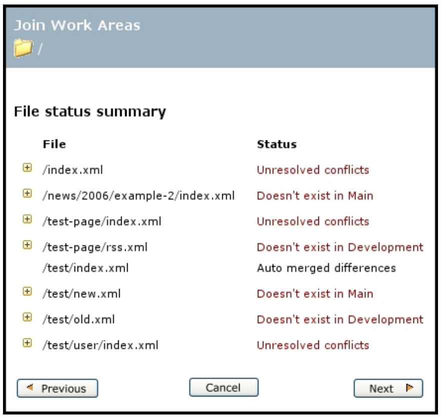
Add to work area

Select to add the file or directory to work area.

Remove from work area

Select to remove the file or directory from work area.

If the wizard finds several files with conflicts, each has to be resolved before continuing to the final step in the wizard. To resolve a conflict, click on the arrow to the left of the file name.


Join Work Areas

/

File status summary

	File	Status
⊞	/index.xml	Unresolved conflicts
⊞	/news/2006/example-2/index.xml	Doesn't exist in Main
⊞	/test-page/index.xml	Unresolved conflicts
⊞	/test-page/rss.xml	Doesn't exist in Development
	/test/index.xml	Auto merged differences
⊞	/test/new.xml	Doesn't exist in Main
⊞	/test/old.xml	Doesn't exist in Development
⊞	/test/user/index.xml	Unresolved conflicts

Joining several files

The image above shows examples of different problems that may occur when joining several files from the two different work areas Main and Test.

10.2.6 Navigation Help

To help the user navigate the wizard, words and different colors have been used to represent different aspects:

Red

Represents a conflict. It is found in the text, to highlight a conflict and to the left in the wizard.

Green

Represents the repository of your current work area.

Blue

Represents the repository of the work area you have chosen to join with.

Gray

Represents old settings or data. When defined as old it is not considered to be a conflict, only a difference between the files, the data has been changed in one file but not in the other.

Old

The word Old is set beside an unselected radio button. It is always represented by gray color.

Del

The word Del is set beside an unselected radio button. Del implies that the data only exists in one file.

Radio buttons

A preselected radio button implies that this is the probable choice a user will make. Select the radio button in front of data you want to keep and the Del button in front of data you want to remove. Data with only an unselected radio button and no Del button in front will be deleted.

By selecting radio buttons the conflicts will be resolved and data removed or added to the work area. The handling of conflicts and differences between files is very similar to the update procedure. The Update page contains a few examples.

10.2.7 After Solving

To complete the join operation, all conflicts must be resolved in the join wizard. When this is done, click on Next to enter the final step in the wizard. Enter a log/commit message to complete the operation, then click on Ok to finish the operation.

Before selecting Ok to finish the operation it is very important that you are sure of what you are doing, and that all choices in the wizard have been made correctly. After all, joining work areas is an important operation and it is hard to undo the operation, especially if you have joined several files. To undo a join operation you have to manually revert each file or delete the unwanted files or directories. To revert a file, focus on it and select the Log button under the Files tab.

11 Workflow

Please see the Workflow section in the Insite Editor manual for an introduction to the workflow module and a detailed description of the workflow panel in the sidebar. Please note that the Insite Editor wizard for viewing workflow status is available in the Content Editor under the “Files” tab in the “View” menu.

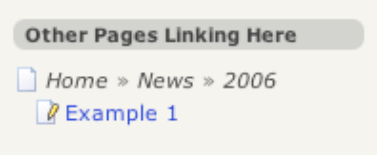
Administrators interested in learning about defining and managing workflow process definitions should read the Workflow section in the Roxen CMS Administrator manual.

12 Link Management

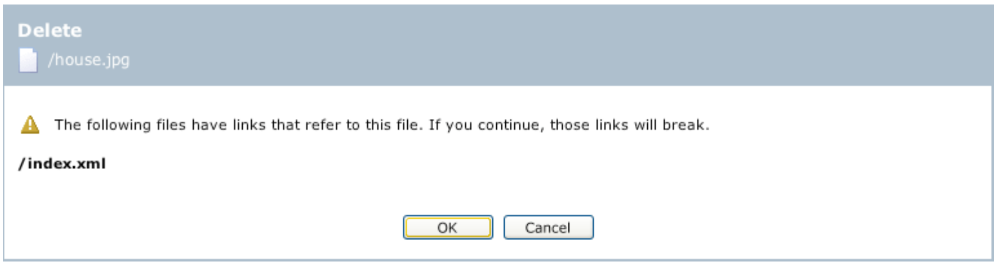
We recommend reading the “Link Management” chapter in the Roxen CMS Administrator manual for a detailed explanation of unique file identifiers and their behavior in Roxen CMS. That documentation also describes various Content Editor wizards that are related to the link management concept.

12.1 User interface

The Sidebar panel lists pages linking to the current page as illustrated here:



In a fashion similar to the Insite Editor toolbar the “Delete" wizard in the “Files” tab will also warn for deletions that break existing links:



The following files have links that refer to this file. If you continue, those links will break.

/index.xml

12.2 Developer API

The link management functions can be broken down in two categories:

- Querying the database to resolve a UUID-enabled link into its corresponding path.
- Getting the UUID-styled link for a given path.

If the caller tries to generate a UUID-styled link for input that is not a valid file path in the repository the reply will be the input string in unmodified form. This means that the caller doesn't have to filter out input such as `http://.../` or `mailto:...` that can appear mixed with paths in XML content.

Please see the RXML/XSLT reference manual for full details of the following API functions or tags. You will find instructions on how to adapt custom editor components to become aware of link management in the “Editor Components” section of the System Developer (Pike) manual.

12.2.1 XSLT templates

Roxen CMS provides two XPath extensions for working with link management:

- **`rxml:resolve-permlink()`**

- **`rxml:get-permlink()`**

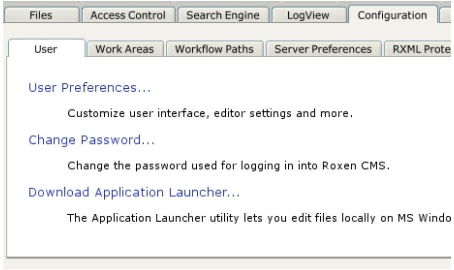
12.2.2 RXML code

RXML developers have link manager functionality similar to XSLT with the addition of an `<emit>` source:

- **`<sb-resolve-permanent-link/>`**
- **`<sb-get-permanent-link/>`**
- **`<emit#sb-referring-pages>`**

13 Customizing the SiteBuilder Editor

It is possible to personalize the SiteBuilder Editor to some extent. The settings are found on the Configuration tab page by clicking on the User preferences button.



User Preferences...

Customize user interface, editor settings and more.

Change Password...

Change the password used for logging in into Roxen CMS.

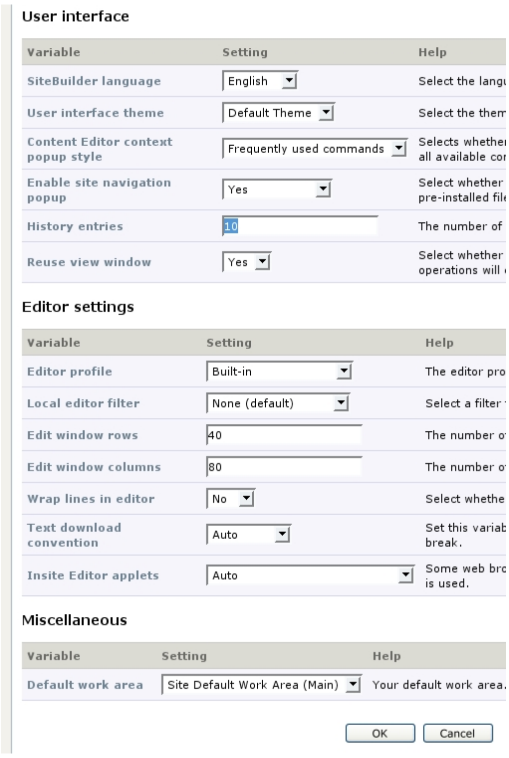
Download Application Launcher...

The Application Launcher utility lets you edit files locally on MS Windo

Configuration view

13.1 User Preferences

User preferences are the user's own personalized settings of the SiteBuilder Editor.



User interface

Variable	Setting	Help
SiteBuilder language	English	Select the langu
User interface theme	Default Theme	Select the them
Content Editor context popup style	Frequently used commands	Selects whethe all available cor
Enable site navigation popup	Yes	Select whether pre-installed fil
History entries	10	The number of
Reuse view window	Yes	Select whether operations will

Editor settings

Variable	Setting	Help
Editor profile	Built-in	The editor pro
Local editor filter	None (default)	Select a filter
Edit window rows	40	The number o
Edit window columns	80	The number o
Wrap lines in editor	No	Select whethe
Text download convention	Auto	Set this variab break.
Insite Editor applets	Auto	Some web bro is used.

Miscellaneous

Variable	Setting	Help
Default work area	Site Default Work Area (Main)	Your default work area.

User preferences

13.1.1 User Interface

SiteBuilder language

Select the language that is to be used in the SiteBuilder interface.

SiteBuilder Editor menu style

Selects the style of SiteBuilder Editor menus.

SiteBuilder Editor context popup style

Selects whether the context popup menus in file listings should contain most frequently used commands only or all available commands.

Enable Javascript

Controls the JavaScript features in the SiteBuilder Editor, such as the pop-up menus, opening a new browser window when viewing a file etc. Default is Yes, i.e. the JavaScript functions activated.

Enable site navigation popup

Select whether the site navigation popup should be available in the SiteBuilder Editor or not.

History entries

The number of entries the history list in the file browser should have. 5 is default.

Reuse view window

Select whether you want to reuse the same window for all view/view source operations. Otherwise, all view operations will open a fresh browser window.

13.1.2 Editor settings

Editor profile

The editor profile controls which editor will be started, the built-in editor or a local editor. The profile can contain different settings for different file types. The editor profiles are always customized for each SiteBuilder installation, usually to reflect the editor software available at each site.

The profile Builtin will always be available. It uses the built-in web-based editor.

Local editor filter

Select a filter for your local XML/HTML editor. Currently, filters are available for Microsoft Word and Netscape Composer. The filter will adapt the contents of XML or HTML files edited in Microsoft Word or Netscape Composer, so the server recognizes them as proper XML or HTML files.

Edit window rows

The number of rows the built-in web-based editor will use.

Edit window columns

The number of columns the built-in web-based editor will use.

Text download convention

This convention controls what type of line breaks will be used in text and HTML files. Is used when a text or HTML file is downloaded, fetched through ftp or sent to a local editor. It is important to set this correct to be able to work with any editor on the local computer.

Insite Editor applets

The Insite Editor interface normally uses Java applets for user-friendly editing of HTML text and tables. However, some web browsers do not implement Java and JavaScript to the necessary extent for the applets to work correctly. Select Auto to have the server decide if the browser is compatible; if not, the applets are replaced by a plain text input field. Select either Always on or Always off to force applets on or off regardless of the web browser type.

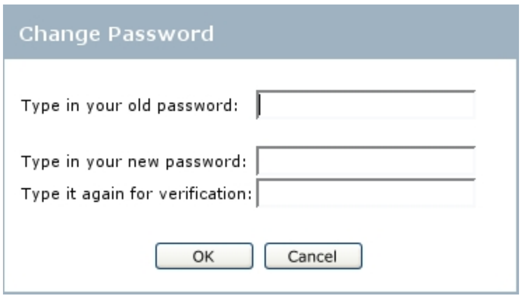
13.1.3 Miscellaneous

Default work area

The work area that will be used when the user first connects to the SiteBuilder Editor. By default this is Site Default Work Area, which is a work area configured by the administrator.

13.1.4 Change Password

The Change password button allows you to change your password via a Change password page. Write the new password twice and confirm the change by clicking Ok. It is also possible to cut-and-paste an encrypted password directly into the Encrypted text field.



Change Password

Type in your old password:

Type in your new password:

Type it again for verification:

Change password

14 Application Launcher

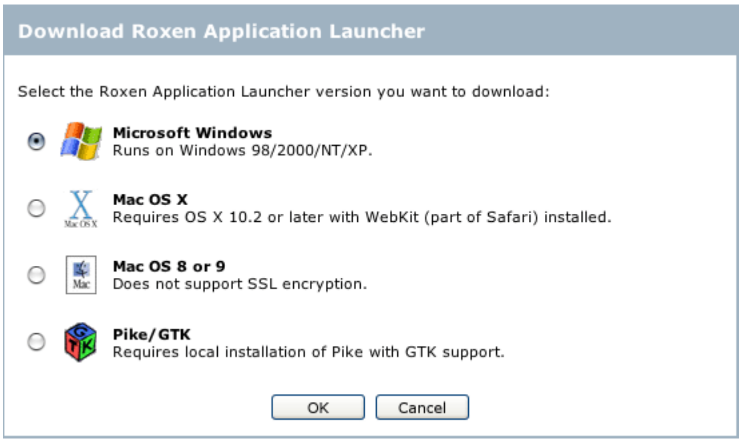
Using a program running on the user's computer or on a network server is usually the best way to edit files. The user can use the HTML editor, drawing program or word processor she is most familiar with.

Application Launcher allows anyone to use their favourite application when editing files using Roxen CMS.

To start an application on a local computer or network a small program, the Application Launcher, has to be installed. This program is the necessary glue between the web browser and the other programs installed on the user's computer or network servers. The Application Launcher helps the computer launch the correct program associated with the file the user wishes to edit.

14.1 Installing Application Launcher on your computer

Go to the Configuration tab and click on "Download Application Launcher..." link to get the following wizard:



Download Roxen Application Launcher

Select the Roxen Application Launcher version you want to download:

● **Microsoft Windows**
Runs on Windows 98/2000/NT/XP.

○ **Mac OS X**
Requires OS X 10.2 or later with WebKit (part of Safari) installed.

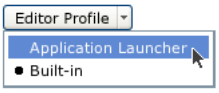
○ **Mac OS 8 or 9**
Does not support SSL encryption.

○ **Pike/GTK**
Requires local installation of Pike with GTK support.

Select the operating system you are using and click "OK" to start the download. Installation and usage instructions differ for each platform so please see the sections below for details.

14.2 Activating Application Launcher for editing files

To enable editing files through the Application Launcher, you need to select an Editor profile that instructs Roxen CMS to send files to your local computer. This is easily done using the popup menu in the top part of the SiteBuilder Editor interface.



Once this is done you can initiate file editing as usual by selecting the normal edit commands ("Edit Contents" in the Edit popup menu).

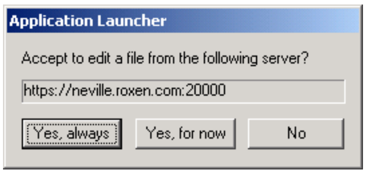
Some file types are excepted from the Application Launcher. This includes menu files, component-based XML files built in the Insite Editor and a few more. The exceptions can be configured by going to the Configuration tab and selecting the Server Preferences sub-tab and opening the Editor Profiles... wizard.

14.3 Application Launcher on Windows

If you wish to reach the Application Launcher interface without editing a file, it can be started from the Programs menu.

The first time a file is edited the Application Launcher will ask some questions.

Trust this server?



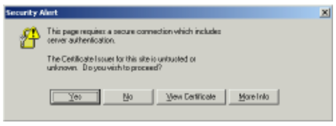
Application Launcher

Accept to edit a file from the following server?

https://neville.roxen.com:20000

Application Launcher will ask if the server is trusted enough to allow editing of files from it. If the No button is clicked, the operation is aborted. The next time a file from that server is to be edited, the Application Launcher will ask the question again. If Yes, always is clicked the server is added to the list of trusted servers found in the Application Launcher Settings tab. This server is now always trusted and this message won't be shown again. Yes, for now allows editing of this particular file. The next time a file from the same server is requested the message will be shown again.

Security alert



Security Alert

This page requires a secure connection which includes server authentication.

The Certificate Issuer for this site is untrusted or unknown. Do you wish to proceed?

If the Roxen CMS server doesn't have a valid signed SSL-certificate from VeriSign/Thawte a standard Windows security alert message will be shown. It asks whether this site can be trusted or not. By clicking the Yes button the application will start with the file in it. If this message is shown the administrator should be notified about it.

If a file with an unknown content-type is edited, Application Launcher will launch the Open with dialogue box and ask what application the file should be associated and opened with.

14.3.1 Troubleshooting

If something goes wrong during the installation or when running the Application Launcher this small troubleshooting guide might help:

Why do I get the error message: 'Application Launcher: denied'?

This problem is most likely a cookie problem. The cookie for the file might be broken or incorrectly set due to the user not having enough privileges to edit it.

The problem might also occur when a Cookie HTTP authentication module hasn't been activated for the Work area. These settings can be found in the SiteBuilder

Editor / Access Control / Resources tab. Click on the Work area that is used in the SiteBuilder Editor, e.g. Work area: Main. The correct Work area is specified in the right top corner below the username. Add Cookie HTTP authentication from the list of module(s) underneath the Active authentication modules heading. If many Work areas are used with the Application Launcher, all should have active Cookie HTTP authentication modules.

Application Launcher won't load any *.gif files, why?

The most probable cause is that the server doesn't allow this particular file to be downloaded. However, there exists a setting in the Content Editor / Configurations / File types tab for this purpose. Make the file downloadable again by clicking on the appropriate content-type and changing Download: No to "Yes". If you can't find a File types button in the Configurations tab, you don't have enough privileges to change this setting. Ask your server administrator to correct the problem.

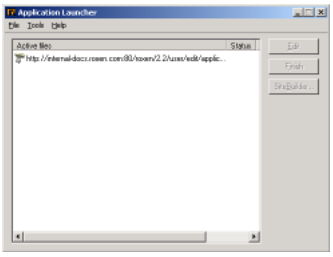
Why do I get the error message: 'download error: server timeout'?

Is "Cookie HTTP Authentication" enabled for each Work Area? If the repository was upgraded from 2.0 this will be missing and needs to be added manually. Ask the Roxen CMS administrator to check the server configuration. Within the Access Control -> Permissions tab inside SiteBuilder click on each Work Area protection class and add the "Cookie HTTP authentication" from the popup menu under the section titled "Active authentication modules".

When the right application is running, editing and saving the document may be done as normal for Windows applications.

The next time a file with the same extension from the same server address is edited, no questions will be asked. Each time a file with a new content-type is edited, the Application Launcher will ask which application should be used to edit that file.

All changes made to a document are saved in a unique temporary directory. Each document has its own directory. The Application Launcher polls the directory through Windows and sends the document changes to the SiteBuilder Editor as long as the Application Launcher is running. The changes are only sent when a "Save" operation is done inside the application.



Active files	Status
http://internal-docs.roxen.com:80/roxen/2.2/user/edit/applic...	

Through the Application Launcher interface all files being edited can be managed. The interface also provides settings for associations between content-types and applications. Menus provide easy access to all available functions.

Also, the two most frequently used functions, Edit and Finish has been given buttons to promote even faster file management without cluttering the interface too much.

14.3.2 The File Menu

Exit

Exits the Application Launcher.

Finish / Finish All

When done editing a file it is not necessary to remove it from the Application Launcher interface, but it might be a good thing to do it anyway else the interface will get cluttered with files. By selecting a file and clicking Finish it will be removed from the interface.

All files can be removed by clicking Finish All.

Performing these operations has consequences that should be taken into consideration.

- Any file removed will also have their own unique directories removed from the Application Launcher's temporary directory. If the user hasn't saved the file in the application it will be lost as the Application Launcher no longer has any information about it.
- The temporary directory will remain if the Application Launcher is shut down by mistake/accident. If this happens, remember to save the file after the Application Launcher has been restarted. Otherwise the new content will be lost if a Finish operation is performed later on.

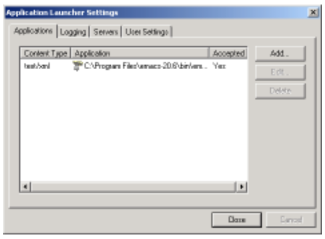
14.3.3 The Tools Menu

Edit

When clicking Edit after selecting a file, the Application Launcher will download a new version of the file from the SiteBuilder. The current version in the temporary directory will be overwritten which means that if the file has been locally altered without Application Launcher running, that information will be lost.

Applications

Brings up the Settings view which contains all associations between content-types and applications.



Content Type	Application	Accepted
text/xml	C:\Program Files\emacs-20.6\bin\em...	Yes

Add

The Add button is used to associate new content-types with an application.

This function is mostly usable for administrators when installing new Application Launchers for users. Before new users get access to the Application Launcher the administrator creates all associations to those applications deemed necessary.

Edit

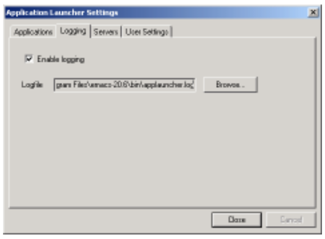
The Edit button is used to change the association between an existing content-type and its application.

The Accept checkbox specifies whether it's permitted or not to edit files with this content-type. This behavior may be necessary if for instance no appropriate application is available or if the user doesn't want to edit a file with this content-type by mistake.

Delete

Removes a content-type.

Logging

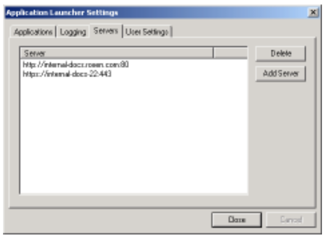


☑ Enable logging

Logfile: gram Files\emacs-20.6\bin\applauncher.log

This view enables logging of the traffic between the server and the Application Launcher. Logging is mostly used for debugging purposes if something is wrong. Specify a log file, e.g. applauncher.log, and a directory where to save it.

Servers

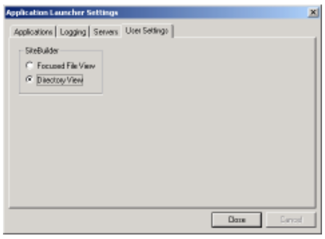


Server
http://internal-docs.roxen.com:80
https://internal-docs-22:443

This view contains all the Roxen CMS servers that are allowed to send files to the Application Launcher. New trusted servers may be added by clicking the Add Server button. Servers no longer trusted may be deleted by clicking Delete.

Trusted servers are those that the administrator or the user find secure enough to allow files to be sent between the users computer and the Roxen CMS server.

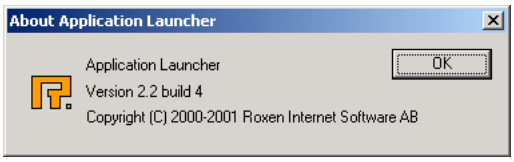
User settings



SiteBuilder
○ Focused File View
● Directory View

In this view it is possible to choose between Focused file view and Directory view of SiteBuilder.

The Help Menu

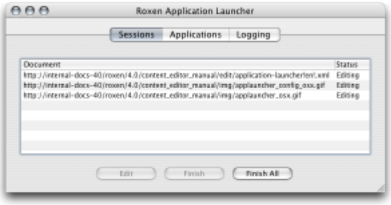


Application Launcher
Version 2.2 build 4
Copyright (C) 2000-2001 Roxen Internet Software AB

14.4 Application Launcher on Mac OS X

To install the program, double-click the disk image you have downloaded. Copy the application to a directory on your hard disk. If you are running Mac OS 8.x or 9.x you also need to start the Application Launcher once to give it a chance to register itself with your web browsers. This last step is not necessary in Mac OS X.

The main window of the Application Launcher on Mac OS X has three tabs.



Document	Status
http://internal-docs-40/roxen/4.0/content_editor_manual/edit/application-launcher/en/.xml	Editing
http://internal-docs-40/roxen/4.0/content_editor_manual/img/applauncher_config_osx.gif	Editing
http://internal-docs-40/roxen/4.0/content_editor_manual/img/applauncher_osx.gif	Editing

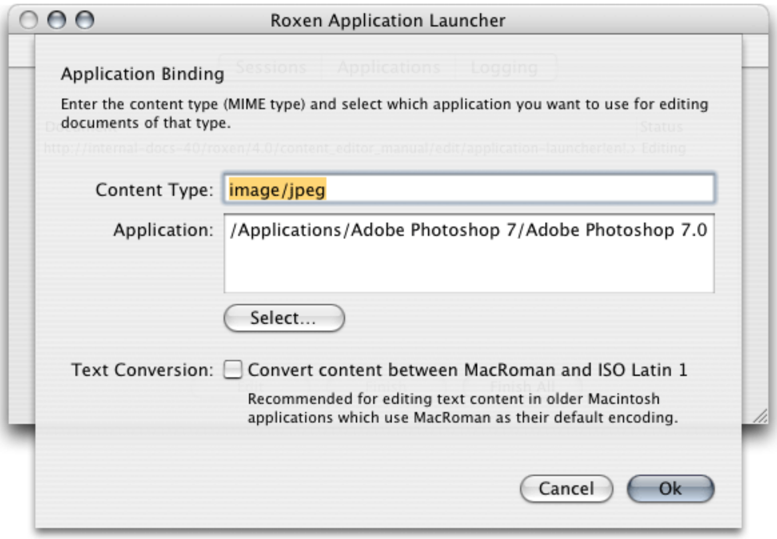
Sessions

The list displays all active editing sessions. When one or more sessions are selected the buttons at the bottom of the window lets you open the documents or end the editing sessions. The "Status" column shows whether the document is being downloaded, edited or uploaded.

Applications

This tab contains the known associations from document content types used in Roxen CMS to local applications that can handle them.

Every time you edit a file type which the Application Launcher doesn't recognize it will inform you that you need to associate the file type with a local program. Use the file selector window to locate a suitable application.



Application Binding

Enter the content type (MIME type) and select which application you want to use for editing documents of that type.

Content Type: image/jpeg

Application: /Applications/Adobe Photoshop 7/Adobe Photoshop 7.0

Text Conversion: □ Convert content between MacRoman and ISO Latin 1

Recommended for editing text content in older Macintosh applications which use MacRoman as their default encoding.

There is a Mac-specific option to perform text encoding conversion. Many text editors expect data in MacRoman encoding while the data on the server may be ISO Latin 1 (ISO 8859-1). By checking the box for this option the Application Launcher will recode the file contents as needed when files are downloaded and uploaded. Do not use this option for binary files such as proprietary document formats, images, multimedia files etc.

Logging

> All downloads and uploads can be logged to a text file. This is useful for troubleshooting purposes. To display the current log, click the button "Open log file in TextEdit".

In Mac OS 8.x and 9.x the procedure for assigning local applications is a bit different. Most importantly you are given a choice of a Mac-specific document type to map the server-side content type to; this is needed since applications normally don't look at file extensions. For example, Adobe Photoshop is capable of opening a wide variety of image formats so you will need to select the corresponding Mac-based file type code from the popup menu – GIF for image/gif, JPEG for image/jpeg and 8BIM for image/x-Photoshop and so on.

14.5 Application Launcher on Unix

By clicking on the Edit button the appropriate local program is started with the correct file loaded. The user can edit and save normally. In case a HTML file is edited the user will still have to press the View button to see the end result after the template has been added.

When the right application is running, editing and saving the document may be done as normal for UNIX applications.

Using the Application Launcher in a UNIX environment is slightly different from using it in a Windows environment.

When a file has been saved in its application and committed to the version control system it is important to close the file in the local editor after the commit, as it will no longer be possible to save the file again. Once a commit has been performed, Roxen CMS will no longer be able to find it. The reason for this is that the file being edited is only available to the local editor while SiteBuilder thinks the user is editing it, or rather while the file exists in the user's edit area. As the user commits the changes, SiteBuilder assumes that the user is no longer interested in editing that particular file and removes it from the user's edit area.

In case the user continues to edit a file that has been committed she must save the file on the local hard disk and use the Upload new file or Upload contents button to upload the changes. It is also possible to press the Edit button after committing the changes, in which case SiteBuilder will start the local program again.

The user must always press the Edit button in the SiteBuilder Editor to start editing a file. It is not possible to use the Open function of the local program, since the file will likely not exist in the users edit area yet. Nor is it possible to rename a file by using the Save As functions in the local program. SiteBuilder will not find the new file. It is however possible to use Save As to save the file to the local hard disk and then use the Upload new file button in the content editor to upload the new file to SiteBuilder.